ESSA TR ERL 177-SDL 15

A UNITED STATES DEPARTMENT OF COMMERCE PUBLICATION

ESSA Technical Report ERL 177-SDL 15

U.S. DEPARTMENT OF COMMERCE
Environmental Science Services Administration
Research Laboratories

SIXBIT
A Generalized Reaction Kinetics Program

G. W. ADAMS

L. R. MEGILL

BOULDER, COLO.
AUGUST 1970

ESSA RESEARCH LABORATORIES

The mission of the Research Laboratories is to study the oceans, inland waters, the lower and upper atmosphere, the space environment, and the earth, in search of the understanding needed to provide more useful services in improving man's prospects for survival as influenced by the physical environment. Laboratories contributing to these studies are:

Earth Sciences Laboratories: Geomagnetism, seismology, geodesy, and related earth sciences; earthquake processes, internal structure and accurate figure of the Earth, and distribution of the Earth's mass.

Atlantic Oceanographic and Meteorological Laboratories: Oceanography, with emphasis on the geology and geophysics of ocean basins, oceanic processes, sea-air interactions, hurricane research, and weather modification (Miami, Florida).

Pacific Oceanographic Laboratories: Oceanography; geology and geophysics of the Pacific Basin and margins; oceanic processes and dynamics; tsunami generation, propagation, modification, detection, and monitoring (Seattle, Washington).

Atmospheric Physics and Chemistry Laboratory: Cloud physics and precipitation; chemical composition and nucleating substances in the lower atmosphere; and laboratory and field experiments toward developing feasible methods of weather modification.

Air Resources Laboratories: Diffusion, transport, and dissipation of atmospheric contaminants; development of methods for prediction and control of atmospheric pollution (Silver Spring, Maryland).

Geophysical Fluid Dynamics Laboratory: Dynamics and physics of geophysical fluid systems; development of a theoretical basis, through mathematical modeling and computer simulation, for the behavior and properties of the atmosphere and the oceans (Princeton, New Jersey).

National Severe Storms Laboratory: Tornadoes, squall lines, thunderstorms, and other severe local convective phenomena toward achieving improved methods of forecasting, detecting, and providing advance warnings (Norman, Oklahoma).

Space Disturbances Laboratory: Nature, behavior, and mechanisms of space disturbances; development and use of techniques for continuous monitoring and early detection and reporting of important disturbances.

Aeronomy Laboratory: Theoretical, laboratory, rocket, and satellite studies of the physical and chemical processes controlling the ionosphere and exosphere of the earth and other planets.

Wave Propagation Laboratory: Development of new methods for remote sensing of the geophysical environment; special emphasis on propagation of sound waves, and electromagnetic waves at millimeter, infrared, and optical frequencies.

Institute for Telecommunication Sciences: Central federal agency for research and services in propagation of radio waves, radio properties of the earth and its atmosphere, nature of radio noise and interference, information transmission and antennas, and methods for the more effective use of the radio spectrum for telecommunications.

Research Flight Facility: Outfits and operates aircraft specially instrumented for research; and meets needs of ESSA and other groups for environmental measurements for aircraft (Miami, Florida).

ENVIRONMENTAL SCIENCE SERVICES ADMINISTRATION

BOULDER, COLORADO 80302

U. S. DEPARTMENT OF COMMERCE
Maurice H. Stans, Secretary

ENVIRONMENTAL SCIENCE SERVICES ADMINISTRATION
Robert M. White, Administrator
RESEARCH LABORATORIES
Wilmot N. Hess, Director

ESSA TECHNICAL REPORT ERL 177-SDL 15

SIXBIT
A Generalized Reaction Kinetics Program

G. W. ADAMS

L. R. MEGILL

SPACE DISTURBANCES LABORATORY
BOULDER, COLORADO
August 1970

For sale by the Superintendent of Documents, U. S. Government Printing Office, Washington, D. C. 20402
Price $1.00

TABLE OF CONTENTS

SIXBIT - A GENERALIZED REACTION KINETICS PROGRAM

G. W. Adams and L. R. Megill

A FORTRAN-IV computer program, SIXBIT, has been written as a general reaction-handling tool. This report describes the use of SIXBIT as a "black-box" program, and describes the program in detail.

Key Words: atmosphere, chemistry, computer, differential equations, program, reactions.

1. INTRODUCTION

In the past few years a number of computer programs designed to solve sets of coupled, time-dependent differential equations, describing the behavior of a set of reactants, have been written. See, for example, Keneshea [1962, 1963, 1967] and Hunt [1966]. The justification for writing yet another of these programs is as follows.

Past programs have been relatively inflexible, requiring rather major reprogramming to vary the set of reactants. In addition, it is often difficult to determine the major process operating in a complicated system. SIXBIT attempts to resolve these two difficulties.

The first objective is answered by subroutine ALGEBRA, which takes a series of cards describing the reactions and writes the appropriate set of differential equations. Once a basic set of reactions is given to the program, adding or deleting one or two reactions is a trivial task.

The second objective is more difficult to answer. An attempt to get out the pertinent information is made by printing out a table giving the major production and destruction mechanisms for each species.

These two features make the program a useful tool for ionospheric research and perhaps for some laboratory investigations. Of necessity,

a number of approximations have been made in handling the solar flux in the atmosphere. The succeeding pages specify those approximations. In addition, no diffusive processes have been included. There are, of course, a significant number of problems in which this omission will make the results invalid.

Extensive measures have been taken both to maintain the accuracy and to keep track of a parameter describing the accuracy of the calculations. The program automatically adjusts the step size as needed to keep the accuracy up. This allows for a great increase in speed, since we find that the step size may change from as short as a few microseconds during twilight to steps of the order of an hour in the middle of the day or night. The resulting program has turned out to be quite fast.

Another feature of the program is that the sun can be "held" in a single position (e.g., noon) while the program is run to steady state. This allows starting values to be obtained so that diurnal variations may be calculated without a steady drift in value of species that have time constants long with respect to a day.

The program is written in FORTRAN IV as used at the Boulder Laboratories CDC 3800. This language allows extensive indexing of indices which is not allowed on many FORTRAN IV systems. However, at least some versions of FORTRAN V allow this flexibility. No extensive measures have been taken to trap for overflows and underflows, and attempts to use this program on machines allowing exponent ranges of ±38 may result in difficulties

The program as presently dimensioned will take 50 reactions, 20 reactants, and 20 heights. These limitations are not fundamental and can easily be extended by modifying dimension statements. The emphasis in this work, however, is to concentrate on the physical processes operating in a system and it is felt that more reactions than this may be exceedingly difficult to interpret.

This report is in two sections. Section 2 describes the use of the program as a "black box." This involves the use of the program without reference to its internal workings. Section 3 describes the program in detail sufficient for the person who may want to modify it.

2. USING SIXBIT

2.1 Data Input

(Appendix A is the listing of one of the data decks actually used for the calculations described in Section 3.)

A. The first input card contains a single integer; 0 or 1. A 0 tells the program to take all the input data from the data cards. A 1 picks up the species concentrations from a magnetic tape, where they were written out in a previous run, but gets the rest of the data from cards. This restart capability is particularly useful on computer systems with restrictive running times.

B. The second card also contains a 0 to 1. A 1 tells the program to make microfilm plots of the profiles; 0 tells it not to.

C. The next three cards contain an alpha-numeric note to yourself so you can remember what you are doing. These three cards will be printed out as the first output of the program.

D. Reactions. Write down on paper your complete reaction set, and make a list of all the species involved. Number the species consecutively. 1 and 2 are not used as species numbers; these are preempted by the program. 3 is O_2 and 4 is O_3. From 5 on, there is no restriction on the species identities. If your reaction set uses O_2 and/or O_3, use these numbers for them. Otherwise, start your numbering with "5". In either case, O_2 and O_3 profiles will be printed out. Now, rewrite your reaction set in terms of the species numbers, making one list for the regular reactions and a second list for the photoreactions. Each reaction must appear in the form of 3 body → 3 body, so anywhere you have blanks, enter a "1". As an example, consider the reaction set

$$(1)\quad A + B \rightarrow C + D \qquad k = 10^{-10}\ cm^3/sec$$

$$(2)\quad C + D \rightarrow A + B \qquad k = 10^{-15}\ cm^3/sec\ .$$

Assuming that neither O_2 nor O_3 is part of this set, we would number the species A = 5, B = 6, C = 7, D = 8. Then the reactions would be

(1) 05 + 06 + 01 → 07 + 08 + 01

(2) 07 + 08 + 01 → 05 + 06 + 01 .

Now express each rate coefficient in the form

$$k = k_o \left(\frac{T}{A}\right)^n e^{B/T}$$

where k_o, A, n, and B are constants, and T is temperature. (The temperature is entered later.) If the rate coefficients are just constants, as in the example, you would have for the first reaction $k_o = 10^{-10}$, A = 1.0, n = 0, B = 0. Note that A must not = 0, since it is used as a divisor. However, with n = 0, any positive value of A will do equally well. The (non-photo-) reactions can now be card-punched, one reaction (with rate coefficient) to a card. There can be any number of reactions up to a maximum of 50. (This number is limited only by the dimension size in the program or the machine memory size and is not fundamental.) In order for the program to know when all the reactions have been read in, the last card in this set must have a "99" punched in columns 1 and 2, with the rest of the card blank. The reaction set will be printed out when the program is run.

```
FORMAT (3(I2,1X),1X,3(I2,1X),6X,2(E8.2,2X),2(E9.2,2X))
SAMPLE ( / IS EDGE OF CARD)
/05_06_01__07_08_01_______1.00E-10__1.00E+00__+0.00E+00__+0.00E+00
/07_08_01__05_06_01_______1.00E-15__1.00E+00__+0.00E+00__+0.00E+00
```

E. Photoreactions. The photoreactions now exist in terms of the species numbers. For these, the reaction cross sections as a function of wavelength must be specified. SIXBIT works with the wavelength region $0 \leq \lambda \leq 12{,}000$Å in 100Å steps, numbered as

1 = 0 - 100Å

2 = 100 - 200Å

3 = 200 - 300Å

. . . .

120 = 11,900 - 12,000Å .

For each reaction, you must specify (a) the number of the first wavelength region with non-zero cross-section, (b) the number of the last region with non-zero cross-section, and (c) the cross-section for each region. To our example reaction set, add the photoreaction

$$C + h\nu \rightarrow A + A \ (07\_01\_01\_\_05\_05\_01)$$

with a cross-section which is $10^{-19}cm^2$ in the region 1850-2400Å and is zero elsewhere. Then we would specify 19 and 24 as the first and last wavelength regions, followed by 5×10^{-20} and five 1×10^{-19}. The first cross-section has been adjusted to give the average cross-section over the region 1800-1900Å. The photoreaction list can contain any number of reactions up to a maximum of 20. It must be terminated with a "99" just like the reaction list was.

```
FORMAT (3(I2,1X),1X,3(I2,1X))
FORMAT (I3,2X,I3)
FORMAT (7(E8.2,2X))
SAMPLE
/07_01_01__05_05_01
/019__024
/5.00E-20__1.00E-19__1.00E-19__1.00E-19__1.00E-19__1.00E-19
```

Notice that both here and in the calculation of the photon attenuation, which is handled separately, 100Å bands are used. This will lead to problems any time the cross-sections have structure, since there is no way to average across such structure and still obey Beer's Law. There are several ways this problem can be handled, but it is left to the user to find a reasonable way to treat any particular case.

F. Altitude Information. Altitudes are in kilometers, with the highest one first. There are two ways to get this information in.

1. Constant height interval. This is denoted by the first number (an integer) being 0. The next three numbers are the highest altitude, the lowest altitude, and the height interval. For example, if you want 30-100 km in 5-km intervals, you punch 0 (to tell it it's constant height interval), then 1.000E+02, 3.000E+01, 5.000E+00.

2. Non-constant height interval. The initial integer is the number of heights to be given, followed by the altitude values in descending order.

For either form of input, a maximum of 20 altitude values is allowed by the present dimension sizes.

```
FORMAT (I2,6(2X,E9.3)) - FIRST CARD
FORMAT (6(E9.3,2X)) - ALL OTHER CARDS
SAMPLE NO. 1 (CONSTANT STEP SIZE)
/00--1.000E+02--3.000E+01--5.000E+00
SAMPLE NO. 2 (VARIABLE STEP SIZE)
/04--1.000E+02--9.500E+01--9.000E+01--8.000E+01
```

G. Temperature Profile. One value of temperature must be given for each altitude, starting with the highest altitude. If none of the rate coefficients has any temperature dependence, then any non-zero values can be entered here.

```
FORMAT (7(E8.2,2X))
SAMPLE
/1.23E+02--1.37E+02--1.46E+02--1.70E+02
```

H. Number of Atoms per Species. The easiest way to maintain accuracy in a calculation like this is to calculate all the densities except one, then determine what that one must be to keep the total number of atoms/cm^3 a constant. Unless the errors get so large that one of the densities goes negative, you will never know anything about the accuracy this way. A better way is to calculate all the densities, then multiply them all by a factor which will keep the total number of atoms/cm^3 a constant. As long as this factor is reasonably close to unity at each step, this approach appears to do well at keeping errors from growing. However, this only controls the total number of atoms, so that oxygen can get converted to nitrogen (for example) and still keep the total number of atoms constant. Therefore, the procedure in SIXBIT is to determine all the different types of atoms which should keep their total numbers/cm^3 constant, such as hydrogen, oxygen, etc. Order these types so that the least abundant one is first and the most abundant one is last. For each species,

the number of each different type of atom is given. Every print-out time, SIXBIT will (1) determine the total number of atoms/cm^3 of the least abundant type, (2) determine the multiplicative factor required to get the total back to the total that was input, (3) multiply those densities which contain that type of atom by this factor, (4) determine the total number of atoms/cm^3 of the next least abundant type, etc. Normally the most abundant type of atom will be the total of all types of atoms. Each time a new factor is determined and used to multiply one of the sets, species #2 will also be multiplied by the factor. Species #2 starts with a value of 1 at all altitudes, and nothing else happens to it. It gets printed out with the other densities, and thus reflects the total amount of "squeezing" that has occurred at each altitude.

```
FORMAT (I2,2X,10(I1,1X))
SAMPLE
/07__2_1_0_0_1_0_4
```

I. Time and Miscellaneous Flags. This consists of one card containing 9 numbers.

1. T, the reaction time relative to the time at which the calculations were started. This is normally input as zero, although it is sometimes convenient to change it when a reaction set is being restarted. Its only essential use for you (although not for the program) is to keep track of time for reaction sets with characteristic times of less than a second.

2. TPRINT, the time interval for which you want the data printed out. This number will automatically be adjusted by the program. If you tell it to print out every second (TPRINT = 1.0), then it will print out the results of the calculations every second for 10 seconds. At this point it will examine the average time steps that the program used in the last print-out intervals. If the

minimum step-size is greater than TPRINT $* 10^{-3}$, then TPRINT will be multiplied by 10. TPRINT will continue to be increased by factors of 10 at 100 seconds, 1000 seconds, etc., up to a maximum (see below), as long as the step-size continues to grow (the normal circumstance for most reaction sets).

3. The solar zenith angle and the photon fluxes are recalculated only at print-out time. For this reason, you do not want TPRINT to get too large. TMAX1 is the maximum value that TPRINT is allowed to have when the sun is above the horizon or when the region of the atmosphere for which you are doing calculations is in total darkness. A reasonable value here is 1 hour (3.6×10^3) for calculations of diurnal variations.

4. TMAX2 is the corresponding upper limit for TPRINT during twilight. One minute is reasonable here, unless you are running to steady state.

5. TQUIT is the stopping time for the program (in seconds). Typically this might be several days (1 day = 8.64×10^4 seconds).

6. IFLAG8 chooses between two methods of solution. A 0 chooses a standard fourth-order Runge-Kutta routine with variable step-size. A 1 chooses a partial integration routine. The choice depends on the type of reaction set you are dealing with. Basically, if the rapidly varying species are minor (relatively low concentration) species in a "sea" of relatively stable major species, then the partial integration routine is usually faster. If the major species are also the rapidly changing species, then the Runge-Kutta routine is usually faster. Most atmospheric problems will therefore run best with the partial integration routine,

while the two-reaction set we have been using for an example will be fastest with Runge-Kutta.

7. IFLAG1 is an integer that controls whether or not species can go into steady state. A 1 here lets species go into steady state when they satisfy the criteria (input later on this card). A 0 does not let species go into steady state.

8. EQUIL is the criterion for letting species go into steady state. If X is the species concentration, then X will go into steady state (assuming it's allowed to go by IFLAG1) if

$$\frac{1}{X}\ \frac{dX}{dt} < \text{EQUIL}$$

10^{-8} is a reasonable value for EQUIL.

9. JOSHUA controls the motion of the sun. If JOSHUA = 0, the sun will go through its normal diurnal variation. If JOSHUA = 1, the sun will remain stationary in the sky. This is used to get steady-state starting values for a reaction set by holding the sun at noon and running to steady state.

```
FORMAT (5(E8.2,2X),2(I1,2X),E8.2,2X,I1)
SAMPLE
/0.00E+00__1.00E+00__3.60E+03__6.00E+01__8.64E+05__0__1__1.00E-08__0
```

J. <u>Local Time Information</u>. This card contains

1. LAT = latitude in degrees
2. LONG = longitude in degrees
3. TMONTH = number of the month
4. TDAY = day of the month

5. THOUR = local hour

6. TMIN = local minute

7. TSEC = local seconds

```
FORMAT (2(E9.2,2X),5(I2,2X))
SAMPLE
/+3.72E+01__+9.25E+01__03__05__23__00__00
```

K. Starting Species Concentration Profiles. The initial value of each species concentration at each altitude can be specified, but not all of them have to be. If O_2 and O_3 (#3 and #4) are not input, internal profiles will be used. However, all altitudes must be in the range 30-200 km to use this feature. Any other species profile not specified will be taken as 5×10^{-6} times the O_2 concentration. (This is somewhat faster than starting the calculations with zeroes.) The species member is entered on a card by itself; the altitude profile is entered on the following card. This set must be terminated with a "99" in the first two columns, as before.

```
FORMAT (I2) - FIRST CARD - SPECIES NUMBER
FORMAT (7(E8.2,2X)) - FOLLOWING CARDS - ALTITUDE PROFILE
SAMPLE
/04
/9.37E+08__4.25E+09__8.34E+09
```

2.2 Program Output
(See Appendix B for a sample)

A. Information Output Only at the Beginning

1. The three alphanumeric cards you input will be printed out.

2. A statement telling how many reactions (excluding photoreactions) and how many species you input is printed. (If

your highest numbered species is 13, then you put in 11 species, since species numbering starts with #3.)

3. A list of the reactions and rate coefficients that you input, numbered sequentially in the order you put them in.

4. The number of photo-reactions input.

5. A list of the photo-reactions, numbered sequentially from the reaction set.

6. A label which says "Starting Values".

7. The local time as you input it.

8. The amount of computer time used up to this point.

9. The reaction time (T) as you input it.

10. The zenith angle of the sun.

11. A table labeled "Reaction Terms" containing, at this point, the rate coefficients for the reactions as a function of altitude.

12. A table labeled "Species Densities" containing the profiles you input, plus the density for species #2, which is 1.0 at this point.

B. Information Output Every Print-Out Time

1. Local time, as above.

2. Total computer time used up to this point.

3. T, the reaction time in seconds relative to the starting point.

4. The solar zenith angle which was used during the last step, therefore the zenith angle appropriate to the previous time, not to the present time.

5. A table labeled "Reaction Terms." For the reaction

$$A + B + C \rightarrow D + E + F$$

with rate coefficient K, the reaction term is

$$K[A]\,[B]\,[C] \quad .$$

In other words, the reaction term is the absolute value of the contribution to the time derivative of species A, B, C, D, E, or F due to this particular reaction. Another interpretation is that it is the number of times per second that this particular reaction is occurring.

Note that the first entry at each altitude in this table is the average step-size in seconds used by the program over the previous print-out interval.

6. A table labeled "Species Densities". Self-explanatory. A minus sign in front of a density indicates that the species is in steady-state. A minus sign in front of an altitude means that all species at that altitude are in steady state.

7. A table labeled "Major Destruction Reactions". For each species at each altitude, a survey is made of all the reactions to see which is the major destroyer. This reaction number is printed out, followed by a slash and then the numbers of the next three destroying reactions, provided that each of the next three is within a factor of 10 of the biggest destroyer. A

"-1" indicates that there are no non-zero destroying reactions for that species. A "-0" indicates that there is no such species.

8. A table labeled "Major Production Reactions". As above.

2.3 Other Things

The program can make microfilm plots of the altitude profiles at each print-out time. Figure 1 shows a sample plot. For local users, the card in subroutine MICROPLT which gives user's name and extension will have to be changed. For other users, subroutine MICROPLT and the CALL MICROPLT statement in the main program (SIXBIT) will have to be removed.

The profiles are written out on tape unit 13 every print-out and then 13 rewinds, so that the last profile is always available. Having the first card input value =1 lets the program read starting profiles, but not the rest of the data, from tape. A dummy read statement is the very first operation in SIXBIT. Therefore, to get started, remove that initial read statement for the first run so that the tape gets labeled, then replace the read statement so that you are always initially reading a labeled tape.

If you find any bugs in the program, please let us know. For people on other machines -- We would appreciate learning what machine you are running on and what changes you find need to be made for your machines. This will be helpful to other people using similar machines.

3. THE INTERNAL WORKINGS OF SIXBIT

3.1 Variable Definitions

(Appendix C is a complete listing of the program.)

A. Indices

I - reaction number, goes from 1 to NREAC.

II - photoreaction number, goes from 1 to NFREAC.

IJZ - reaction rate constant index for reaction I at altitude JZ, goes from 1 to (I)(JZ).

ILOOK - searching index used in PHYSICS, goes from 1 to 4.

ITCH - type-of-atom index, goes from 1 to ITCHXX.

JRK - Runge-Kutta index, goes from 1 to 4.

JZ - altitude number, goes from 1 to JZEND.

JZTAB - altitude index used for tabulated values of O_2 and O_3. Runs from 1 to 87, corresponding to altitudes of 2ØØ-28 km in 2-km steps.

LAMDA - wavelength number, goes from 1 to 12Ø corresponding to wavelength bands Ø-1ØØÅ, 1ØØ-2ØØÅ, etc., up to 11,9ØØ-12ØØØÅ.

M - species number, goes from 1 to NSPEC.

B. Dimensioned Variables (All contained in COMMON /A/)

A(I) - Input, read and used in ALGEBRA, part of the temperature-dependent rate coefficient, K = KØ*(TEMP/A)**N*EXPF(B/TEMP). Should never be zero, since it's used as a divisor.

ARK(JRK) - one of the sets of constants (set in START) used in the Runge-Kutta technique.

B(I) - See A(I).

BRK(JRK) - See ARK.

CRK(JRK) - See ARK.

F(M) - The sum of all gain terms for Y(M), so that the differential equation is written $\frac{dY(M)}{dt}$ = F(M) - G(M)*Y(M) where F(M) $\geq \emptyset$ and G(M) $\geq \emptyset$. Includes both terms from regular reactions and terms from photoreactions.

FF(M) - Same as F(M) evaluated Δt = H later. Used to calculate backwards in time to see if the species densities return to where they started.

FLUX(LAMDA) - The photon flux (photons/cm^2-sec-1ØØ Å) calculated at a particular altitude and zenith angle. Calculated in PRODUC.

FLXINF(LAMDA) - The photon flux (photons/cm^2-sec-1ØØ Å) outside the atmosphere. Tabulated in table 1, plotted in figure 2. Set in a DATA statement in START.

FRATE(II) - The cross-section times the photon flux, summed over wavelength, for the IIth photoreaction. FRATE times the species density Y(LL1(II)) gives the number of times per second that the reaction goes. Calculated in PRODUC, used in WEIRD.

FTERM(II) - FRATE(II)*Y(LL1(II)), one of the photo-terms which gets added up to form the differential equations.

G(M) - The sum of all loss terms for Y(M). See F(M).

GG(M) - G(M) evaluated at t + H. See FF(M).

HBAR(M) - The average step-size over the previous print time. Used in WEIRD to get an initial value for H, and printed out.

H2HOLD(JZ) - The scale height of O_2 at a particular time. Calculated in ZENANG from the O_2 profile; used in PRODUC.

H3HOLD(JZ) - Same as H2HOLD but for O_3.

IATOM(ITCH,M) - The number of atoms of type ITCH per molecule, for species M, used to determine the total number of atoms per cm^3 as an error check. Input.

IGAIN(ILOOK) - The numbers of the reactions which are the dominant gain processes for a particular species at a particular altitude. This gets transferred to IGAINA, B, C, or D. Determined in PHYSICS.

IGAINA(M,JZ) - The reaction which is the biggest producer of species M at altitude JZ. This will be set = -1 if there are no producers (this is possible since PHYSICS does not look at photoreactions), and = Ø if there is no such species. This gets calculated in PHYSICS and is part of the program's output.

IGAINB(M,JZ) - The reaction which is the second biggest producer of species M at altitude JZ, provided it's within a factor of 10 of IGAINA. See IGAINA.

IGAINC(M,JZ) - The third biggest producer of M at JZ. See IGAINB.

IGAIND(M,JZ) - The fourth biggest producer of M at JZ. See IGAINB.

ILOSS(ILOOK) - The dominant loss processes for a particular species at a particular altitude. This gets transferred to ILOSSA, B, C, or D. See IGAIN.

ILOSSA(M,JZ) - The reaction which is the biggest destroyer of species M at altitude JZ. See IGAINA.

ILOSSB(M,JZ) - Loss equivalent of IGAINB.

ILOSSC(M,JZ) - Loss equivalent of IGAINC.

ILOSSD(M,JZ) - Loss equivalent of IGAIND.

INFORM(INFO) - 3∅ words of information to yourself. Three cards of alpha-numeric reminders which get written out at the first of a run. Read in SIXBIT.

JPULL(JZ) - A flag set to 1 is all species are in steady state at altitude JZ; = ∅ if they are not. Calculated in SIXBIT.

K(IJZ) - The rate coefficient for reaction I at altitude JZ. See A.

K∅(I) - The constant (non-temperature-dependent) part of K. See A.

KABS(II,LAMDA) - The input absorption cross-section for the IIth photo-reaction. Read in REACTS, used in PRODUC to calculate FRATE.

KABSN2(LAMDA) - The absorption cross-section for N_2, in cm^2 $(1∅∅Å)^{-1}$. Set by a DATA statement in START. Plotted in figure 3, tablulated in Table 1.

KABSO2(LAMDA) - The absorption cross-section for O_2, in cm^2 $(1∅∅Å)^{-1}$. Tabulated in Table 1; plotted in figure 3. Set by a DATA statement in START; used in PRODUC to calculate FLUX. Taken from Allen [1963].

KABSO3(LAMDA) - The absorption cross-section for O_3, given in Table 1 and figure 4. See KABSO2.

KRK(MJRK) - The four first derivatives for each species calculated for the Runge-Kutta solution.

L1(I) - The number of the first species on the left-hand side of the Ith reaction. Input, read in REACTS.

L2(I) - The second species on the left-hand side of the Ith reaction.

L3(I) - The third species on the left-hand side of the Ith reaction.

LL1(II) - The first species on the left-hand side of the IIth photoreaction.

LL2(II) - The second species on the left-hand side of the IIth photoreaction.

LL3(II) - The second species on the left-hand side of the IIth photoreaction.

N(I) - See A.

NASRAS(M*I)- The Number And Sign of the Reaction Affecting the Species. A long array, divided up into a region for each species. Each region contains the reaction numbers of all reactions which are destroyers of that species, and reaction number + NREAC for all reactions which are producers of that species.

NFTERM(M) - The number of photoreactions which involve the Mth species calculated in ALGEBRA.

NTERM(M) - The number of reactions which involve the Mth species.

NUSRUS(M*II) - The number and sign of the photoreaction affecting the species. See NASRAS.

PHOLD(M,JZ) - An integer array which contains a "1" if species M at altitude JZ is in steady-state; a "Ø" if it's not. This gets set in SIXBIT and is used to flag the printed species density with a minus sign if it's in steady state. The run is terminated when all species are in steady-state; i.e., when PHOLD contains all 1's.

PROHLD(I+II,JZ) - Contains all the TERMs and all the FTERMs; used for print-out.

QOLD(M) - A holding array used in RUNKUT to hold the values of QRK for each step temporarily in case the error is too big and the step has to be repeated.

QRK(M) - One of the elements used in the Runge-Kutta solution.

R1(I) - The first species on the right-hand side of the Ith reaction.

R2(I) - The second species on the right-hand side of the Ith reaction.

R3(I) - The third species on the right-hand side of the Ith reaction.

RR1(II) - The first species on the right-hand side of the IIth photo-reaction.

RR2(II) - The second species on the right-hand side of the IIth photo-reaction.

RR3(II) - The third species on the right-hand side of the IIth photo-reaction.

SUMIN(ITCH,JZ) - The total number of atoms/cm^3 of type ITCH (see IATOM) at altitude JZ which were input. This is calculated in START, then used in SIXBIT to compare with SUM (the current number of atoms/cm^3) and to correct the densities of all the species which contain atoms of type ITCH to force SUM to equal SUMIN.

TABO2(JZTAB) - Tabulated values of O_2 density, given every 2 km from 2ØØ to 28 km. Used to get an O_2 profile if one is not input.

TABO3(JZTAB) - Tabulated O_3 densities. See TABO2.

TEMP(JZ) - The input temperature profile, in degrees Kelvin. Used as a divisor, therefore, must not contain zero values.

TERM(I) - The quantity K*Y(L1)*Y(L2)*Y(L3) for each reaction. Each $\frac{dY}{dt}$ is calculated as a sum of a set of TERMs. TERM(I) contains TERM for the Ith reaction; TERM(I+NREAC) also contains TERM for the Ith reaction.

TERMW(I) - Used in WEIRD, = TERM(I) calculated at Δt = H later than TERM. Used to calculate backwards in time to check on the error and adjust the step size.

Y(M) - The density (molecules/cm^3) of species M.

YHOLD(M,JZ) - The array of Y(M) at all altitudes.

YLAST(M,JZ) - The values in YHOLD at the previous print time. Used to compare to YHOLD to see what has gone into steady-state.

YNEW(M) - The values of Y at Δt = H, calculated in WEIRD. Used to start the backwards calculation.

YOLD(M) - A holding array for Y used in RUNKUT. Used to restart the step if the error is too big. See QOLD.

YPRINT(M,JZ) - The same as YHOLD, except that, if a species is in steady-state, then the value is negative. This is the array that gets printed out.

YSTART(M,JZ) - The starting values of YHOLD. Used to determine if the species are in steady-state.

Y3SUN(JZ) - The O_2 profile used in ZENANG to calculate the O_2 scale heights (H2HOLD). This will be the same profile as in YHOLD if the solar zenith angle is less than 9Ø°. If it's greater than 9Ø°, then Y3SUN will have the last O_2 profile before the zenith angle exceeded 9Ø°.

Y4SUN(JZ) - The O_3 profile used in ZENANG. See Y3SUN.

Z(JZ) - The altitudes, in kilometers, at which the calculations are being done. Highest one first. A maximum of 30 are allowed.

ZPRINT(JZ) - Same as Z except that the value is negative if all species at that altitude are in steady state. Used when printing out the species densities.

C. <u>Non-Indexed Variables</u> (Contained in COMMON /B/)

ALPHA - The solar zenith angle, in degrees, calculated in ZENANG.

EQUIL - The input criterion for deciding when a species is in steady-state. If $\frac{\Delta Y}{Y\Delta t}$ < EQUIL, then the species is in steady-state.

H - The calculation step-size, in seconds, used in WEIRD and RUNKUT.

HCOUNT - A counter used to count the number of steps taken by RUNKUT or WEIRD at each altitude. Used to calculate HBAR.

IFLAG1 - Input. = 1 lets species go into steady-state; = Ø it does not.

IFLAG2 - Used by RUNKUT and WEIRD to know when it's time to return to the main program.

IFLAG3 - Used by ZENANG to know when it's the first time through so that starting values can be printed out.

IFLAG4 - Used to adjust the print time TPRINT.

IFLAG5 - Used to adjust TPRINT. See IFLAG4.

IFLAG6 - Used in START to tell if an O_2 profile was input or if one should be constructed from TABO2.

IFLAG7 - Used to tell if an O_3 profile was input. See IFLAG6.

IFLAG8 - Input. Ø to use RUNKUT: 1 to use WEIRD.

IFLAG9 - Set to 1 if an error in the data is found. Run is terminated when it's = 1.

IJZX - = JZEND-JZMAIN. Used as part of the index for K.

IJ99 - = the number 99, written on magnetic tape at end of YHOLD(M,JZ) array.

IUNIT - The number of the tape unit where the program is to get the start-profiles. 6Ø if it's a fresh start; 13 if it's a restart.

IXX - Dummy index.

JCNTRL - Carriage control character used in one FORMAT statement in HANDLE so that the spacing on the starting values comes out right. Set in ZENANG.

JFLAG2 - Input. Ø for a fresh start, 1 for restart. See IUNIT.

JFLAG6 - Input. Ø says that fixed spacing heights are being given. If not Ø, then = the number of altitudes being given explicitly.

JOSHUA - Input. 1 holds the sun fixed in the sky; Ø lets the sun move naturally.

JZEND - The highest altitude index, equals the number of different altitudes being used.

JZMAIN - The altitude index of the main altitude loop in SIXBIT. Used by several subroutines.

LAT - The latitude, in degrees, of the place to do the calculations for.

LONG - The longitude, in degrees, of the place to do the calculations for. (Not used).

LX1 - first specie on the left side of the reaction.

NFREAC - The number of photoreactions input.

NREAC - The number of regular reactions input.

NSPEC - The number of species input, including #1 and #2. Actually the highest numbered species input.

PI - 3.14159.

RX1 - first specie on the right side of the reaction.

T - Input initially. The time, in seconds, for which calculations have been done. Normally input as zero. Normally of no concern unless the reaction set is of interest on a time scale of less than a second, in which case T is the only thing that will keep track of the calculation time.

TDAY - Input. The number of the day. Changed as T increases and is printed out.

THOUR - Input. The hour of the day. See TDAY.

TMAX1 - Input. The maximum value which TPRINT is allowed to have during non-twilight conditions.

TMAX2 - Input. The maximum value which TPRINT is allowed to have during twilight conditions.

TMIN - Input. The minute of the hour. See TDAY.

TMONTH - Input. The month of the year. See TDAY.

TOLD - The value that T had at last print-out.

TPRINT - Input. The time between print-outs. This gets increased by X1Ø periodically if the smallest average step-size is within a factor of $1\emptyset^3$ of TPRINT, up to a maximum of TMAX1 or TMAX2.

TQUIT - Input. Quitting time in seconds. The program will terminate when T = TQUIT.

TSEC - Input. The second of the minute. See TDAY.

ZBOTUM - Input if JFLAG6 = Ø. The lowest altitude to be used, in km.

ZSTEP - Input if JFLAG6 = Ø. The altitude interval to be used, in km.

ZTOP - Input if JFLAG6 = Ø. The highest altitude to be used, in km.

3.2 How SIXBIT Sets Up the Differential Equations

A reaction set is read in numerical form, for example

Reaction No.	Reaction	Rate Coefficient Terms
1	∅3 ∅4 ∅1 ∅5 ∅6 ∅1	1.∅E-1∅ 1.∅E∅ ∅.∅ ∅.∅
2	∅5 ∅6 ∅1 ∅3 ∅4 ∅1	1.∅E-15 1.∅E∅ ∅.∅ ∅.∅
3	etc.	

For reaction #1, the first species number on the left-hand side (∅3) is L1(1), the second (∅4) is L2(1), the third (∅1) is L3(1), the first species number on the right (∅5) is R1(1), the second (∅6) is R2(1), and the third (∅1) is R3(1). The total number of reactions read in is NREAC, and the highest species number is NSPEC. Having read in the reactions, the program then determines which reactions affect each species, and whether or not each reaction is a destroyer or a producer of that species. This is accomplished by building a long one-dimensional array of reaction numbers. The first NREAC locations are for species #3, the second NREAC locations are for species #4, etc. If a reaction is a destroyer of the species, then the reaction number itself is entered in the table. If the reaction is a producer, then the reaction number + NREAC is entered. The table of reaction numbers is called NASRAS (Number and Sign of the Reactions Affecting the Species). If the reaction set consisted of the first two reactions in the example given above, then NASRAS would be set up as

Location

1) Allocated to species #3. Length 2, because there are
2) 2 reactions in the set

3) Allocated to species #4
4)

5) Species #5
6)

7) Species #6
8)

and would be filled as follows:

Location	Contents	Explanation
1	1	Reaction #1 is a destroyer of species #1.
2	4	Reaction #2 is a producer of species #1 and has had NREAC(=2) added to it.
3	1	
4	4	
5	3	
6	2	
7	3	
8	2	

The number of reaction numbers entered for each species is counted by NTERM which is indexed by the species number. In the example just given, NTERM(3) = NTERM(4) = NTERM(5) = NTERM(6) = 2, since there have been 2 reaction numbers entered for each species.

Given the arrays NASRAS and NTERM, the differential equation for each species can be formed as

$$\frac{dY}{dt} = \sum_i \text{TERM}_i$$

where i takes on all reaction numbers found in NASRAS, TERM = $K_i \, Y_{L1_i} \, Y_{L2_i} \, Y_{L3_i}$ and TERM is taken positive or negative depending on whether the reaction is a producer or destroyer of Y.

In this way, SIXBIT reads the reactions and sets up the differential equations. Since the TERMs must be recalculated every step during the solution of the equation, the only things that exist permanently in the program are the reactions, NASRAS, and the current values of the species densities.

3.3 How SIXBIT Handles Sunlight and Photochemistry

One of the inputs to SIXBIT is latitude (LAT), longitude (LONG), date and time (TMONTH, TDAY, THOUR, TMIN, TSEC). The declination of the sun is calculated from

```
DEC = 23.445*SINF((2.Ø*PI*(30.4*TMONTH+TDAY-112.0))/365.Ø)
```

which is just a sine fit to the variation of the declination of the sun over a year, with the approximation that all months have 30.4 days.

The hour angle of the sun is calculated as

```
TZEN = 15.Ø*ABSF(FLOATF(TSEC+60*(TMIN+60*THOUR))-43200)/3600
```

and the zenith angle is then calculated as

```
ALPHA = (1/RAD)*ACOSF(SINF(LAT*RAD)*SINF(DEC*RAD)+COSF(LAT*RAD)*COSF
(DEC*RAD)*COSF(TZEN*RAD))  .
```

Once the program has ALPHA, it calculates the solar spectrum for the altitude where it's doing the calculations. To do this, the program carries the solar spectrum at the top of the atmosphere. This is carried as FLXINF(LAMDA), where FLXINF (Flux at infinity) is the flux in photons/cm^2-sec-100 Å and LAMDA is the wavelength index. The spectrum goes from 0 to 12,000 Å in 100 Å steps, so LAMDA goes from 1 to 120.

Wavelength Region	LAMDA
0-100 Å	1
100-200 Å	2
200-300 Å	3
. . . .	. .
11,900-12,000 Å	120

The solar spectrum is shown in figure 2.

The solar spectrum at the altitude of interest is calculated as

$$\phi(\lambda) = \phi_\infty(\lambda) \exp \left\{ - \sigma_{O_2}(\lambda)[O_2]H_{O_2}F_{O_2} - \sigma_{O_3}[O_3]H_{O_3}F_{O_3} \right\}$$

where $\sigma_{O_2}(\lambda)$ is the absorption cross-section of O_2 as a function of wavelength, $\sigma_{O_3}(\lambda)$ is the absorption cross-section of O_3, $[O_2]$ and $[O_3]$ are the O_2 and O_3 concentrations, H_{O_2} and H_{O_3} are the O_2 and O_3 scale heights at the altitude of interest, and F_{O_2} and F_{O_3} are factors which play the same role that sec α would in a plane-parallel atmosphere. $\sigma_{O_2}(\lambda)$ and $\sigma_{O_3}(\lambda)$ are carried by the program. The values used are shown in figures 3 and 4. $[O_2]$ and $[O_3]$ are the densities calculated at each point in time by the program, except that, when the sun goes down below the horizon, the O_2 and O_3 profiles at 90° are held and used until the sun comes back up to 90° again. The physical justification for this is that, for $\alpha > 90^\circ$, the absorption is controlled primarily by the density at the point of closest approach to the earth of the sun's rays, which is the point at which $\alpha = 90^\circ$.

H_{O_2} and H_{O_3} are calculated from the existing profiles of O_2 and O_3. F_{O_2} and F_{O_3} are determined from equations given by Swider [1964].

The flux at a given altitude is determined entirely from absorption coefficients carried by the program. The photo-production (or loss) rates of particular species are determined from the calculated fluxes and from absorption coefficients which must be read into the program. Absorption coefficients, like the flux values, are handled in 100-Å bands. Values must therefore be entered so that the average over the 100Å band is correct.

A special case exists for Lyα (LAMDA = 13). Since most of the energy is at 1215Å,the cross section of O_2 suitable to 1215Å is used in the cross-section value.

All the photo calculations are repeated every print-out time, so print-out times must be controlled so that the change in solar zenith angle is not excessive. (See TMAX1 and TMAX2 in the input data.)

Solar spectra at a variety of altitudes, calculated by SIXBIT, are shown in Figures 5, 6, 7, and 8.

3.4 Methods of Solution of the Time-Dependent Differential Equations

There are two methods of solution within SIXBIT, with selection made via a 0/1 flag in the input data. 0 is a Runge-Kutta solution; 1 is a partial integration technique. In general the partial integration technique works best if the rapidly changing species are minor species which are imbedded in a relatively stable sea of major species (which covers most atmospheric problems) - Runge-Kutta would be best for the simple 2-reaction set we have been using as an example.

A. The Runge-Kutta Routine, RUNKUT

This has been taken with no changes from Romanelli [1964].

B. The Partial Integration Routine, WEIRD

The equation for any species can be written as

$$\frac{dY_m}{dt} = \sum_{i \neq m} K_i \, Y_{1i} \, Y_{2i} \, Y_{3i} \; - \; Y_m \sum_{j \neq m} K_j \, Y_{1j} \, Y_{2j}$$

which says that Y_m can be factored out of every loss term and never appears in the gain terms. (This is an approximation if species m appears twice or more on the left-hand side of a reaction and not at all on the right-hand side.) Now rewrite this as

$$\frac{dY_m}{dt} = f_m - g_m Y_m$$

where f_m and g_m can be functions of all the Y's except Y_m.

Now, taking f_m and g_m to be constant over a step in time, this can be integrated directly to give

$$Y_m(H) = \left\{ Y_m(0) - \frac{f_m}{g_m} \right\} e^{-g_m H} + \frac{f_m}{g_m}$$

where f_m and g_m are evaluated at $t=0$. The accuracy of a step is checked by starting with $Y_m(H)$ and calculating backwards in time ($t=H$) to get $Y_m^B(0)$, which should be the same as $Y_m(0)$. Agreement to 1 part in 10^3 is satisfactory here in most cases. If the agreement is not this good, the step-size is halved and the calculation repeated. If the agreement is satisfactory, then the step forward is recalculated using the average f and g over the interval.

Any typical method for solving differential equations (such as Runge-Kutta) makes two approximations; that most quantities stay constant over a step and that the integral can be approximated by a Taylor's series. The partial integration method merely recognizes that there is no reason to approximate the integral since it can be evaluated as it stands.

3.5 Through the Program in Detail

In this section we will go through the program in the logical (working) order, with special attention to areas which may have to be changed to run on other machines.

There are no separate dimension statements. All dimensioned variables are in COMMON/A/, which is carried by the main program and all the subroutines. COMMON/B/ carries all the non-indexed variables needed in more than 1 subroutine. Type statements (REAL and INTEGER) have been used to allow more reasonable variable names.

The comment cards carry enough information to get the data in properly.

READ (13,997) ICRUD - This is a dummy read which just insures that the program will always read from unit 13 (which holds the profiles for restart purposes) before it writes on 13. This is necessary on our system to avoid changing control cards between runs.

CALL REACTS - Subroutine REACTS reads in the reactions, counts the reactions and the species numbers, and reads and counts the photoreactions. The following describes REACTS.

DO 1∅ I1=1,51

. . .

1∅ CONTINUE - This loop reads in the reactions until it finds a 99 in columns 1 and 2, at which point it jumps to statement 2∅. If it does not find a 99 in the first 51 reactions, it will print an error message, set IFLAG9 = 1, return to the main program, and terminate. This loop also searches out the largest species number existing in the reactions.

2∅ NREAC = I1-1 - The number of reactions is I1-1 since it was on I1 when it found the 99, and that is not a reaction.

NSPEC = NSPEC-2 - This gets the actual number of species input, not the highest numbered species. (This assumes that you have used all the numbers between 3 and NSPEC. If NSPEC = 1Ø but 5 is never used as a species number, no harm is done, but the program thinks 5 is a real species.)

DO 3Ø I2=1.21

. . .

3Ø CONTINUE - This reads in the photoreactions; same game as the regular reactions.

RETURN to SIXBIT

CALL HEIGHTS - This is a trivial subroutine which sets up the altitudes.

IF(JFLAG6.EQ.Ø) GO TO 1ØØ - JFLAG6=Ø says that you have given the program the highest altitude, the lowest altitude, and the step size (in kilometers). JFLAG6≠0 says you have given it JFLAG6 altitudes, the highest one first.

RETURN to SIXBIT

CALL ALGEBRA - ALBEGRA calculates the rate coefficient array K and sets up the arrays NASRAS, NUSRUS, NTERM, and NFTERM.

DO 26Ø I2=1,NREAC

. . .

26Ø CONTINUE - This loop sets up NASRAS and NTERM. The first part, down to statement 2ØØ, just eliminates species which appear on both sides of the reaction.

RETURN to SIXBIT

CALL START - This does most of the initializing needed.

The first DATA statement contains the three arrays of constants needed by RUNKUT (ARK, BRK, and CRK) and initializes several arrays. HBAR is set to 1 since WEIRD uses HBAR to get a beginning step-size. The first two species in YHOLD are set to 1, since these are the unit density species that the program uses.

The second and third DATA statements contain FLXINF, the solar photon flux at infinity in photons/cm^2-sec-100Å, starting with the 0-100Å band (LAMDA = 1).

The fourth and fifth DATA statements contain KABSN2, the absorption coefficient for N_2 in cm ^{2}, starting with the value for the 0-100Å band. This absorption coefficient is not used at present. If you ever want to do F-region calculations, UV absorption by N_2 may become important.

The sixth and seventh DATA statements contain KABSO3, the absorption coefficient for O_3 in cm ^{2}.

The eighth and ninth contain KABSO2, the absorption coefficient for O_2 in cm ^{2}.

The tenth contains TABO2, the tabulated O_2 densities in molecules/cm^3, given from 200 km to 28 km in 2 km steps.

The eleventh DATA statement contains TABO3, the tabulated O_3 densities, just like TABO2.

DO 1Ø M1=3,NSPEC

. . .

1Ø CONTINUE - reads in species numbers and the number of atoms/molecule of the different types. Note that one card must be present for every species, but that they do not have to be in order.

READ (6Ø,994) . . .

READ (6Ø,993) . . . - the latitude and calendar information gets read in here even though it is not needed until subroutine ZENANG. This is done so that the starting concentration profiles, which do get read in START, can be the last cards in the data deck. By having the data arranged in this way, JFLAG2 can be set to pick up the starting profiles from cards or magnetic tape without having to rearrange the data deck.

DO 2Ø M3=2,NSPEC

. . .

2Ø CONTINUE - This loop reads in the starting profiles. It keeps reading until it finds a "99" for a species number, so the loop is set to go from 2 to NSPEC. This way, even if you give it a profile for every species, it will try to read one more and thus find the "99". Each time it reads a species number, it looks to see if that species is #3 or #4 (O_2 or O_3). That way, if O_2 and/or O_3 profiles are not input, it can generate its own from TABO2 and/or TABO3, since it has to have O_2 and O_3 profiles to do the sunlight calculations. It also set YNEW(M4) = 2.Ø for each species for which it has profiles, so that any species not specified can be given a profile later of 5×10^{-6} times the O_2 profile. This is just to avoid having zeroes for profiles, since the solutions are often quite slow when they have to work with true zeroes rather than just small numbers.

PRINT 996

IFLAG9=1

GO TO 9000 - if the previous loop never found a "99" to indicate the end of the profiles, it will print an error message, return to SIXBIT, and terminate the run.

3Ø IF(IUNIT.EQ.13)REWIND 13 - if the loop found a "99", it will come here and rewind tape unit 13 if it read 13 to get the profiles. IF(IFLAG6.EQ.1 etc. - if both O_2 and O_3 profiles were input, it's happy and will go to 4Ø. If the O_2 and/or O_3 profiles were not input, then IF(Z(1).LE.200.0 etc. - the altitude range must lie within 30-200 km or else the program cannot construct O_2 and/or O_3 profiles internally. If it is not, it will terminate.

4Ø DO 5Ø JZ3=1,JZEND

. . .

5Ø CONTINUE - This loop sets up the O_2 profiles to be used for the sunlight calculations (Y3SUN) from the tabulated values,

regardless of whether or not a profile was input. If the calculations are being started in daytime, then this profile will be replaced after the first step by the calculated profile. If the calculations are being started at night, then this avoids having the sunrise period being controlled by a nighttime profile. I2 and I1 are the index values of the TABO2 values which are at altitudes that are required for the calculations. This loop also puts the O_2 profile derived from TABO2 into YHOLD if an O_2 profile was not input (i.e., if IFLAG7 = 0).

DO 7Ø JZ4=1,JZEND

. . .

7Ø CONTINUE - This does the same thing for the O_3 profiles that the previous loop did for the O_2 profiles.

8Ø DO 85 M6=5,NSPEC

. . .

85 CONTINUE - This loop makes use of the flag values contained in TNEW (set in the "DO 2Ø" read loop) to put in profile = 5×10^{-6} times the O_2 profile if the profile was not input.

DO 11Ø ITCH=1,1Ø

. . .

11Ø CONTINUE - This loop calculates the total number of atoms/cm^3 of each different type in the input data. CHECK is used to find out how many different types of atoms have been declared. ITCHXX is set to the total number of different types of atoms (actually to the first value of ITCH which is followed by CHECK < 1.Ø).

. . .

RETURN TO SIXBIT

Everything down to this point will be passed through by the program once per run. The statements after this are in a big loop which will be passed through many times.

1∅ CALL ZENANG - This subroutine calculates the zenith angle of the sun (ALPHA), prints out the starting values that were input, and calculates the scale heights of O_2 and O_3 for all the altitudes.

IF(IFLAG3.NE.∅) GO TO 1∅ - IFLAG3 was set to zero in the first DATA statement in START. The first time through ZENANG, ALPHA is calculated, the starting values are printed out, and IFLAG3 is set to 1. The rest of the time, since IFLAG3 is not = 0, printing the starting values is omitted, and the calculation of ALPHA is skipped if the sun is being held fixed in the sky.

DEC = ... - This calculates the declination of the sun in degrees. This equation is a sine fit to the annual variation of the solar declination, and carries the approximation that every month has 30.4 days.

PRINT 998 - This puts the "STARTING VALUES" label on the print-out.

JCNTRL = ∅ - This is a carriage control character that will keep the print-out from skipping to the top of the next page, so that the "STARTING VALUES" label will be in the proper place.

THOLD = TPRINT

TPRINT = ∅ - This will keep time from being advanced by the print-out routine as is normally done.

THOLD will let TPRINT be recovered after the print-out.

TZEN = . . . - This calculates the hour angle of the sun, which is measured in degrees (1 hour = 15 degrees) relative to noon.

ALPHA = . . . - This calculates the zenith angle of the sun, in degrees away from vertical.

CALL HANDLE - This is the subroutine that does all the print-out.

TX = . . .

. . .

TSEC = TX - This section adds TPRINT onto the previous calendar time to get the new calendar time. Note that TX, TDAY, THOUR, TMIN, TSEC are all integers by virtue of the type statement.

ITSEC = TSEC + 100

. . .

ITDAY = TDAY + 100 - on the CDC 3800, an I format which is too small drops digits from the left. Therefore, 104 printed in an I2 format comes out an 04. This lets 8 AM on June 1 be printed out as 06/01 08/00/00 rather than as _6/_1 _8/_0/_0.

ITIME = KLOCK(1)

TIME = ITIME/1000 - KLOCK is a system subroutine that reads a clock in the CDC 3800 in milliseconds, relative to the start of the program.

The rest of this subroutine is a straightforward print-out of the reaction terms, the species densities, the major destruction reactions, and the major production reactions.

Right after statement #4∅ (CONTINUE) is IF(JCNTRL.EQ.∅) GO TO 9000. Since the major destruction and production reactions are not determined until after the calculation has been advanced through a step, their print-out is skipped the first time through. (Note that JCNTRL is used as a carriage control character in FORMAT 999, used in the first print statement.)

RETURN to ZENANG

JCNTRL = 1 - This carriage control character is now set to 1 and will stay that way throughout the rest of the run.

TPRINT = THOLD - The true value of TPRINT is recovered.

GO TO 1∅∅ - This will skip by the section which starts with statement #1∅. This section, reached by the first statement in the program (IF IFLAG3 · · ·) is just the calculation of ALPHA without all the extra statements required the first time through.

1ØØ IF(ALPHA·GT·9Ø.Ø+SQRTF(Z(1))) GO TO 9000 - The equation $\alpha_s = 90 + \sqrt{Z_s}$, where α_s is the zenith angle of the sun in degrees and Z_s is the height of the shadow of the solid earth in kilometers, is a useful approximation. This statement says to skip all the calculations about solar fluxes if the height of the shadow is above the highest altitude of interest.

IF(ALPHA.GT.9Ø.Ø) GO TO 3ØØ - if $\alpha < 90^o$, the following loop will take the O_2 and O_3 profiles to be used for the solar flux attenuation calculations (Y3SUN and Y4SUN) as the current calculated profiles (YHOLD). If $\alpha > 90^o$, it wants to use the profiles that existed at 90^o (or the closest it got to 90^o), so it skips the replacement loop.

DO 2ØØ JZ1=1JZEND

. . .

2ØØ CONTINUE - This just puts the O_2 and O_3 profile into Y3SUN and Y4SUN.

The rest of ZENANG calculates the scale heights of O_2 and O_3 at the altitudes of interest, since this is what PRODUC needs to calculate the attenuation of the solar photon flux. It does this by fitting the top two points in each profile with an exponential, and determining from this the total amounts of O_2 and O_3 that lie above the top altitude. At each altitude, it then determines the total amount of O_2 and O_3 from there to the top by doing a simple sum, adds on the amount determined to lie above the top altitude, and then determines what the scale heights have to be at that point for the integrated densities above that point to come out right.

The exponential form is

$$Y(Z_1) = Y(Z_2)\ e^{-(Z_1-Z_2)/H}$$

so that, if Z_1 is the top altitude and Z_2 is the next-to-the-top, then

$$H = (Z_2-Z_1)/\ln(Y(Z_1)/Y(Z_2)) \quad .$$

The total number of molecules/cm^2 above Z_1 is

$$X = \int_{Z = Z_1}^{Z = \infty} Y(Z_1)\, e^{-(Z - Z_1)/H}\, dz$$

which gives

$$X = HY(Z_1) \quad .$$

Now back to the program.

300 IF(Y3SUN(2).GT.Y3SUN(1)) GO TO 310 - If the top two points on the profile indicate that the density is increasing with altitude, then the exponential extrapolation of the profile to infinity will give an infinite number of molecules. Therefore, if the density is increasing, the program will print out a message (PRINT 997), set IFLAG = 1, return to SIXBIT, and terminate. If the density is decreasing, it goes to statement 310.

310 IF(Y3SUN(1).GT.0.0) GO TO 320 - This avoids a division by zero in determining the scale height for O_2. If the top O_2 density is zero, then

H2HOLD = 1.0 - The scale height is taken as 1 and it jumps to 330. Otherwise it goes to

320 H2HOLD(1) = (Z(2)-Z(11)/LOGF(Y3SUN(1)/Y3SUN(2)) which is just the equation for the scale height given earlier.

330 IF(Y4SUN(2).GT.Y4SUN(1)) GO TO 340.

350 H3HOLD(1) = . . . - This is the same as the preceding block, except for O_3 instead of O_2.

400 ADD3 = Y3SUN(1)*H2HOLD(1)

ADD4 = Y4SUN(1)*H3HOLD(1) - ADD3 and ADD4 are the integrated densities of O_2 and O_3. These statements, using the equation for X given before, initialize the sums.

DO 410 JZ2=2,JZEND - This starts the integration.

ADD3=ADD3+(Y3SUN(JZ2-1)+Y3SUN(JZ2))*(Z(JZ2-1)-Z(JZ2))/2.Ø

ADD4 = . . . - These take the total numbers of molecules/cm^2 between two altitudes as the linear average of the end-points times the distance between the two altitudes, and adds them onto the sum.

H2HOLD(JZ2)=ADD3/Y3SUN(JZ2)

H3HOLD(JZ2) = . . . - This uses the equation $X = HY(Z_1)$ to determine the effective scale height at JZ2.

41Ø CONTINUE

RETURN to SIXBIT

DO 1ØØ JZMAIN=1,JZEND - Here begins the calculation proper. JZMAIN is in COMMON/B/ and will be used in several places to know what altitude is being treated.

IJZX = JZEND - JZMAIN - This is part of a compound index that will be needed in RUNKUT or WEIRD.

DO 3Ø M1=1,NSPEC

. . .

3Ø CONTINUE - This transfers the species densities at this altitude to a singly-dimensioned array, since it is faster to use a single index than to use a double index.

CALL PRODUC - This is the subroutine that uses the scale heights calculated in ZENANG to determine the solar flux at Z(JZMAIN), and then determines the FTERM's for the photoreactions from this flux.

CALL Q9EXUN - This is a system subroutine which suspends the underflow detection, and simply replaces an underflow (a number less than $\sim 10^{-308}$) by zero without writing an error message.

IF(NFREAC.EQ.0) GO TO 9000 - If there are no photoreactions, this subroutine will be skipped.

ALPHAR=ALPHA*PI/18Ø.Ø - ALPHAR is ALPHA in radians.

REARTH = 6371.Ø - The radius of the earth in kilometers.

IF(ALPHA.GT.9Ø.Ø) GO TO 1ØØ - This statement and statement 1ØØ divide the calculations into 3 regions; $\alpha < 90^o$, $90 < \alpha < 90 + \sqrt{Z}$, and $\alpha > 90 + \sqrt{Z}$. For $\alpha < 90^o$, the calculation of F02 and F03 is done using equation 53 in Swider [1964]. For $90 < \alpha < 90 + \sqrt{Z}$, F02 and F03 are calculated with equation 47 in Swider. If $\alpha > 90 + \sqrt{Z}$, the photo terms are set to zero with the loop at statement 1000. F02 and F03 can be thought of as sec α modified for the curvature of the earth.

Having calculated F02 and F03 for either $\alpha < 90^o$ or $90 < \alpha < 90 + \sqrt{Z}$, you wind up at

3ØØ DO 31Ø LAMDA1=1,12Ø

. . .

31Ø CONTINUE - This loop calculates the solar photon flux as a function of wavelength, FLUX(LAMDA), using the flux at infinity (FLXINF), which was given in a DATA statement in START, the absorption coefficients for O_2 and O_3 (KABSO2 and KABSO3) which were given in DATA statements in START, the O_2 and O_3 densities (Y3SUN & Y4SUN) which were set up in ZENANG, and F02 and F03 which were just calculated.

DO 33Ø II=1,NFREAC

. . .

33Ø CONTINUE - The rate that a photoreaction of the form $A + h\nu \rightarrow B + C$ goes is

$$\frac{dA}{dt} = [A] \int_0^\infty \sigma(\lambda)\phi(\lambda)d\lambda \quad .$$

In SIXBIT, the integral is called FRATE. This loop just calculates FRATE.

GO TO 9000

9ØØØ CALL R9EXUN - This restores the system underflow detection.

RETURN to SIXBIT

HCOUNT = 0 - This is a counter that will be used later.

IF(IFLAG8.EQ.1) GO TO 4∅ - IFLAG8 is the input flag (0 or 1) that picks the method of solution, either RUNKUT (the Runge-Kutta solution) or WEIRD (the partial integration solution).

RUNKUT:

CALL Q9EXUN - This suspends underflow detection, as in ZENANG.

H=HBAR(JZMAIN)*3.∅ - HBAR was set to 1.∅ in START. Later HBAR will be the average step-size over the previous print interval. This statement assumes that the biggest allowable step is probably bigger than the previous average. H is the step to be used in the calculation.

1∅ TCHECK = T-TOLD+H

. . .

IFLAG2 = 1 - If a step of size H will put the time beyond the time for the next print-out, then H is reduced to get time just to the print time, and IFLAG2 is set to 1. This will return the program to SIXBIT at the end of the step.

2∅ DO 3∅ M1=2, NSPEC

...

3∅ CONTINUE - This holds onto the current values of Y and QRK in case the error is too big in the next step and the calculation must be repeated.

4∅ DO 9∅ JRK=1,4 - This starts the 4 calculations involved in a fourth-order Runge-Kutta solution.

DO 5∅ I1=1,NREAC

. . .

5∅ CONTINUE - This calculates the TERM for each reaction (see section 2).

DO 52 IT1=1,NFREAC

. . .

52 CONTINUE - This calculates the FTERM's for the photoreactions.

JRKX = (JRK-1)*NSPEC - Part of a compound index to be used later for KRK.

DO 7Ø M2=3,NSPEC - This starts the loop that calculates the time derivatives for the species.

MJRK = JRKX+M2 - The rest of the compound index to be used later for KRK.

IIIX=(M2-3)*NREAC - Part of a compound index to be used for NASRAS.

DY=Ø.Ø - DY is the time derivative for species M2. It will be transferred to an indexed variable after all the TERM's have been added on.

NNSTOP=NTERM(M2) - This is the number of TERM's in NASRAS which affect species M2. This statement is made because NTERM has to be the end of a DO loop specification, and indexed variables cannot be used for that.

DO 6Ø NN=1,NNSTOP - This begins the loop to pick the TERM's for M2 out of NASRAS and add them onto DY.

IF(NASRAS(IIIX+NN).LE.NREAC) GO TO 55 - This picks a reaction number from NASRAS and decides if it is a production reaction or a destruction reaction. If it is >NREAC, it is a production reaction, and it goes to

DY=DY+TERM(NASRAS(IIX+NN)), which adds the TERM for that reaction to DY. If the reaction number is ≤NREAC, it goes to

55 DY=DY-TERM(NASRAS(IIIX+NN)). Either way, it winds up at

6Ø CONTINUE, which sends it back to get the next reaction number from NASRAS.

IIFX=(M2-3)*NFREAC

. . .

65 CONTINUE - This section is just like the one before, except that this one picks the FTERM's from NUSRUS (the photoreactions) and adds them to (or subtracts them from) DY.

KRK(MJRK)=DY - This just puts DY into an indexed array.

7Ø CONTINUE - This is the end of the M2 loop.

Now that it has all the time derivatives, it does the loop

DO 8Ø M3=3,NSPEC

. . .

8Ø CONTINUE - which calculates Y at one of four places in the interval H in a standard Runge-Kutta method.

9Ø CONTINUE is the end of the Runge-Kutta loop begun at statement #4Ø.

K5MAX = Ø.Ø.1

. . .

1ØØ CONTINUE - This determines the maximum value (K5MAX) of a ratio of differences of the KRK's, which is a measure of the error. If K5MAX < Ø.1, the step had too big an error and it should be repeated with a smaller step.

IF(K5MAX.LT.Ø.1) GO TO 12Ø - If the error is too big, it cuts the step-size down with

H=H*Ø.Ø8/K5MAX. It sets

IFLAG2=Ø, uses

DO 11Ø M5=2,NSPEC

. . .

11Ø CONTINUE to recover the starting values, and goes back to statement #4Ø to redo the step. If the error was small enough, it winds up at

12Ø T=T+H which advances the time

H=H*Ø.Ø8/K5MAX which gets a value for the next step

HCOUNT=HCOUNT+1 - which is a counter on the number of steps it's taken (this was zeroed in SIXBIT), and uses

IF(IFLAG2.EQ.Ø) GO TO 1Ø to see if it is time to return to SIXBIT or not.

CALL RETURN - Restores the system underflow.

WEIRD

Section 3.4 B gives the general approach used here.

AKRACY=1.ØE-3 - This is the largest allowed difference between the starting value of the species density (Y) and the value back-calculated from T+H (YBACK).

HMIN=1.ØE-6 - is the minimum allowable step-size. H will never be allowed to get smaller than this regardless of the accuracy requirement.

YNEW(1)=1.Ø - This will be needed later.

CALL Q9EXUN - Suspends system underflow detection (see ZENANG or RUNKUT).

H=HBAR(JZMAIN)*3.Ø - as in RUNKUT

IF(H.LT.TPRINT). . .

. . .

IFLAG2=1 - as in RUNKUT

1Ø DO 1ØØ I1=1,NREAC

. . .

2ØØ CONTINUE - Calculation of TERM's and FTERM's as in RUNKUT.

DO 5ØØ M1=3,NSPEC

. . .

5ØØ CONTINUE - This calculates the quantities F and G. This is the same as the calculation of DY in RUNKUT, except that all gain terms are added to F, all loss terms are added to G, and Y is factored out of G.

6ØØ DO 7ØØ M2=3,NSPEC

. . .

7ØØ CONTINUE - This loop calculates the values of Y(YNEW) with a time step H. The equation used is that given in Section 3.4 B, with sub-cases for F and/or YG being effectively zero.

DO 8ØØ I2=1,NREAC

. . .

12ØØ CONTINUE - This section calculates the new TERM's (TERMW), the new phototerms (FTERMW), and the new F's and G's at the point T+H, just as the first part of this subroutine calculated TERM's, FTERM's, F's and G's at the point T.

DELMAX = Ø.Ø

. . .

1400 CONTINUE - This section calculates backwards in time, just as before except with a step-size -H, gets the new values of Y at time T+H-H (YBACK), determines the difference between YBACK and Y(DEL), and keeps track of the maximum value of DEL(DELMAX).

IF(DELMAX.LE.AKRACY) GO TO 1500 - If the maximum error (DELMAX) is bigger than AKRACY, then the step-size is checked with IF(H.LT.1.01*HMIN) GO TO 1500. If the step-size is bigger than the minimum value, then the step is decreased either by a factor of 2 or down to HMIN, whichever is larger. IFLAG2 = 0, and it returns to 600 to repeat the step with a smaller step-size. If either DELMAX < AKRACY or H ~ HMIN, then it winds up at

1500 DO 1700 M6=3,NSPEC

. . .

1700 CONTINUE which just repeats the step forward using the average values of F and G (FBAR and GBAR).

T=T+H

. . .

9000 CALL Q9EXUN - just as in RUNKUT except for the way the step-size is increased.

RETURN TO SIXBIT

Control is now returned to SIXBIT from RUNKUT or WEIRD, whichever was being used. The species densities (Y's) for the altitude Z(JZMAIN) have been advanced from T to T+TPRINT, and HCOUNT contains the number of steps that were required.

50 HBAR(JZMAIN)=TPRINT/HCOUNT - gives the average step-size used over the series of calculations.

IFLAG2 = 0 - reset for the next altitude.

DO 65 ITCH=1,ITCHXX

. . .

70 CONTINUE - This determines the number of atoms of each different type/cm3 at this altitude according to the new set of densities, scales the densities so that the number/cm3 stays constant,

scales Y(2) every time to accumulate the total amount of scaling that has occurred, and puts the densities back into YHOLD.

DO 73 I=1,NREAC

. . .

76 CONTINUE - The TERM's and FTERM's from the last step in RUNKUT or WEIRD are stored in PROHOLD. They will be printed out later.

T=TOLD - T is reset to TOLD for the calculations at the next altitude.

1ØØ CONTINUE - Go back and take the next altitude.

Having completed the JZMAIN loop, all altitudes have been advanced from T to T+TPRINT.

T=T+TPRINT - T was reset to TOLD at the end of the JZMAIN loop, this puts it back to where it should be.

IF(IFLAG1.EQ.Ø) GO TO 15Ø - IFLAG1 was one of the input parameters. If IFLAG1= 0, then it does not make equilibrium checks. If IFLAG1 =1, then it does.

DO 14Ø JZ1=1,JZEND

. . .

14Ø CONTINUE - This determines what has gone into equilibrium. A species is declared to be in equilibrium (1) if both the current species density (YHOLD) and the species density at the last print-out (YLAST) are less than 10^{-50}, i.e., if it looks like it is at zero and plans to stay there, (2) if the current species density is down by 10^5 or more from the starting value (YSTART) and is also less than 1.Ø, i.e., it looks like it is going to zero, (3) if the fractional change per second $(\frac{1}{Y}\frac{\Delta Y}{\Delta t})$ over the last print interval is less than the value of EQUIL specified on the input data. If YHOLD(M,JZ) meets any of these criteria, then is it declared to be in equilibrium, then JPULL(JZ) is set =1 and HBAR(JZ) is set to TQUIT so that it will not interfere with a later section which adjusts TPRINT, the print-out interval.

15∅ DO 17∅ JZ2=1,JZEND

. . .

17∅ CONTINUE - This loop transfers YHOLD to YPRINT and Z to ZPRINT. When it transfers, it uses PHOLD to make YPRINT negative if the species is in equilibrium, and it uses JPULL to make ZPRINT negative if all the species at that altitude are in equilibrium,

CALL PHYSICS - This subroutine determines which reactions are the major producers and destroyers of each species at each altitude. It gives the major producer (or destroyer) of each species, and then any other reactions which are within a factor of 10 of the major one. All the TERM's and FTERM's that PHYSICS needs are in PROHLD, but it must look at either NASRAS or NUSRUS (depending on where it is) and then decide if the reaction it has is a producer or a destroyer.

DO 8∅∅ M1=3,NSPEC - This picks a species number to look at.

IIIX=(M1-3)*NREAC

IIFX=(M1-3)*NFREAC - Part of compound indices that will be needed later.

NNSTOP=NTERM(M1)+NFTERM(M1) - Part of a DO-loop parameter, similar to the ones used in RUNKUT and WEIRD

NTX=NTERM(M1) - Just a renaming since this gets used quite a bit later.

DO 7∅∅ JZ1=1,JZEND - This picks a particular altitude to look at.

DO 1∅∅ ILOOK1=1,4

. . .

1∅∅ CONTINUE - This zeros the two arrays. The next section will make four passes through the reactions. The first pass will determine the biggest destroyer of species M1 at altitude JZ1 and put the reaction number in ILOSS(1), and determine the biggest producer and put that reaction number in IGAIN(1). The second pass will pick up the next biggest producers and destroyers and, if they are within a factor of ten of the biggest ones, put them into ILOSS(2) and IGAIN(2), etc.

DO 6ØØ ILOOK2=1,4 - This starts the four passes described above.

ILMAX=IGMAX=0

TLMAX=TGMAX=Ø.Ø - ILMAX and IGMAX are the reactions which are the biggest destroyer and biggest producer of species M1. TLMAX and TGMAX are the values of the TERM for ILMAX and the TERM for IGMAX.

DO 5ØØ NN=1,NNSTOP - This starts the search through the TERM's and FTERM's.

IF(NN.GT.NTX) GO TO 11Ø - If NN ≤ NTX (which is the same as NTERM(M1)) then it is looking at the regular reactions. If NN ≥ NTX, it is looking at photoreactions.

It gets the reaction with

NX=NASRAS(IIIX+NN) or with

11Ø NX=NUSRUS(IIFX+NN-NTX). In either case, they wind up at

15Ø ILX=ILOOK2-1 which is a parameter that will be needed later.

IF(NN.GT.NTX.AND.NX.GT.NFREAC) GO TO 3ØØ - If NN > NTX, it is looking at photoreactions. If NX > NFREAC, then it is looking at a producer of M1. It thus goes to statement #300.

IF(NN.GE.NTX.AND.NX.GT.NREAC) GO TO 3ØØ - If NN ≤ NTX, it is looking at regular reactions. If NX > NREAC, then it is looking at a producer of M1. It thus goes to statement #300.

DO 2ØØ ILOOK3=1,ILX

. . .

2ØØ CONTINUE - This looks through the list of the destruction reactions already found to see if this reaction has been entered previously. If it has, it jumps to statement #5ØØ.

Otherwise it winds up at

25Ø IF(NN.GT.NTX) GO TO 26Ø - This statement and the nine following (through the third GO TO 5ØØ) compare the TERM (or FTERM) to the biggest TERM found so far (TLMAX). If this TERM (or FTERM) is bigger than TLMAX, then TLMAX is set equal to this TERM, the reaction number (NX) (or NX+NREAC in the case of photoreactions) is stored in ILMAX, and it goes to statement #5ØØ.

3ØØ IF(NN.GT.NTX) GO TO 31Ø

. . .

5ØØ CONTINUE - This section handles production reactions just as destruction reactions were handled in the preceding section. The biggest TERM is stored in TGMAX, and the reaction number is IGMAX.

IF(ILOOK2.EQ.1) GO TO 55Ø - If this is the first time through, it goes to

55Ø ILOSS(1)=ILMAX

. . .

BIGG=TGMAX - Which puts ILMAX into ILOSS(1), IGMAX into IGAIN(1), and holds the TERM values in BIG2 and BIGG for later use. If ILOOK2 is greater than 1, it does

IF(TLMAX.GE.Ø.1*BIG2) . . .

. . .

GO TO 6ØØ - Which enters those reactions into the appropriate ILOSS and IGAIN only if they are within X10 of the maximum values. Having finished the ILOOK2 loop, it goes to

IF(ILOSS(1).EQ.Ø) GO TO 61Ø - If ILOSS(1)=Ø), then there were no loss processes for M1, and -1 will be entered into the final arrays with statement #61Ø. Otherwise, the final arrays (ILOSSA, ILOSSB, ILOSSC, and ILOSSD) are filled with the values of ILOSS+100, which gives leading zeroes as in HANDLE.

62Ø IF(IGAIN(1).EQ.Ø) . . .

. . .

63Ø . . . - Same as above for the gain processes.

7ØØ CONTINUE - Return to get another altitude.

8ØØ CONTINUE - Return to get another species.

RETURN TO SIXBIT

CALL HANDLE - This subroutine, which prints out the TERM's and FTERM's the species densities, and the major production and loss reactions determined in PHYSICS, was discussed earlier.

IF(JJFLAG.NE.1) GO TO 175 - JJFLAG was input as Ø or 1. If it was Ø, the next statement is skipped. If it was 1, it goes through

CALL MICROPLT - This subroutine generates microfilm plots of the species densities. Samples are given in Appendix D. No details of the workings of this subroutine will be given. Anyone not working on the Boulder CDC 3800 will have to delete this subroutine and the call to it. For people using the CDC 3800, note that the name and extension in the first DATA statement must be changed so that we do not get your microfilm.

RETURN TO SIXBIT

175 TOLD=T - TOLD is advanced in preparation for the next set of calculations.

IF(TOLD.GE.TQUIT) GO TO 8ØØ - TQUIT was the time to quit which was input. If T < TQUIT, it goes to the next section, which increases the print interval (TPRINT).

This section works well only if T was input as Ø.Ø initially. In normal operation, the result of this section is to have the print-outs come at T=1, 2, 3, . . ., 9, 1Ø, 2Ø, 3Ø, . . ., 9Ø, 1ØØ, 2ØØ, 3ØØ, . . ., 9ØØ, 1ØØØ, 2ØØØ, etc., which keeps you from being buried in paper while still giving regions of rapid change in sufficient detail to see what is going on.

IFLAG4=IFLAG4+1 - IFLAG4 is a counter which is treated as the second significant digit of T. (This is true if T starts at Ø.Ø and if TPRINT is 1 x 10^n.)

IF(IFLAG4.LT.1Ø) GO TO 188 - The adjustment of TPRINT is done every tenth print-out. When IFLAG4 = 1Ø, then it sets it back to Ø and advances IFLAG5 by 1. IFLAG5 is treated as the first significant digit, so it gets reset to zero when it reaches ten.

HMIN=TPRINT

. . .

18Ø CONTINUE - This determines HMIN, the minimum average step-size (HBAR) over the previous print interval.

IF(HMIN/TPRINT.LT.Ø.ØØ1) GO TO 188 - If more than 1000 steps were required at any altitude during the previous print interval, then

the next section is skipped, i.e., TPRINT is not adjusted. If less than 1000 steps were required, then TPRINT will be adjusted, so it goes to

IF(ALPHA.LT.9∅.∅.OR.ALPHA.GT.9∅.∅+SQRTF(Z(1))) GO TO 182 - If $\alpha < 90$ or $\alpha > 90 + \sqrt{Z}$, then the altitudes of interest are either in total daylight or total darkness. If neither is true, then the region is in twilight. The following section, i.e.,

IF(TPRINT*1∅.∅.LT.TMAX2) GO TO 184

. . .

184 TPRINT=TPRINT*1∅.∅ - Increase TPRINT either by X10 or up to TMAX1 (whichever is smaller) for daylight or dark conditions, or increase TPRINT either by X10 or up to TMAX2 (whichever is smaller) for twilight conditions. (TMAX1 and TMAX2 are the input upper limits for TPRINT). Then

IFLAG4=IFLAG5

IFLAG5=0

GO TO 19∅ - which shifts the identities of the first and second significant digits in T, and goes to statement #19∅.

188 IF(ALPHA.GT.90.AND.ALPHA.LT.9∅.∅+SQRTF(Z(1)).AND.TPRINT.GT.TMAX2) TPRINT=TMAX2 - This statement was reached either when it was not yet time to check on the adjustment of TPRINT (3 statements after #175) or when the average step-size was too small to increase TPRINT (one statement past #18∅). In either case, this statement checks to see if the sun has entered the twilight region, and if it has, decreases TPRINT if need be down to TMAX2.

19∅ DO 21∅ JZ4=1,JZEND

. . .

21∅ CONTINUE - YHOLD is shifted into YLAST so that YLAST will be available at the next print-out for checking on steady-state.

DO 22∅ JZ5=1,JZEND

IF(JPULL(JZ5).EQ.0) GO TO 1∅

22ϕ CONTINUE - JZPULL(JZ5) was set =1 if all the species at that altitudes were at steady-state. If any of the JZPULL's are not =1, then it goes back to statement #1ϕ to start another set of calculations. If all the JZPULL's are =1, then it winds up at

7ϕϕ PRINT 999

GO TO 9ϕϕϕ - Which prints out a statement that "ALL SPECIES AT ALL ALTITUDES ARE NOW IN EQUILIBRIUM" and quits.

4. CONCLUSION

In this report we have given the details of the use of a computer program designed to give solutions of sets of reactions. The main virtue of the program is its generality and the ease with which different reactions can be used.

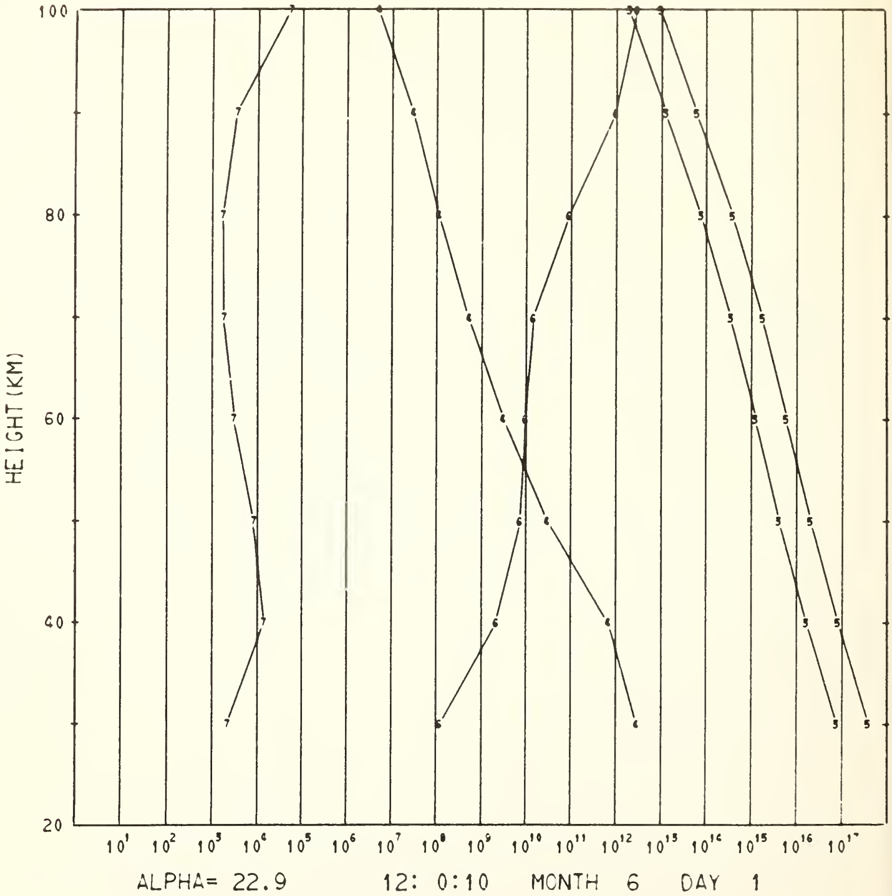

Figure 1. Sample microfilm plot.

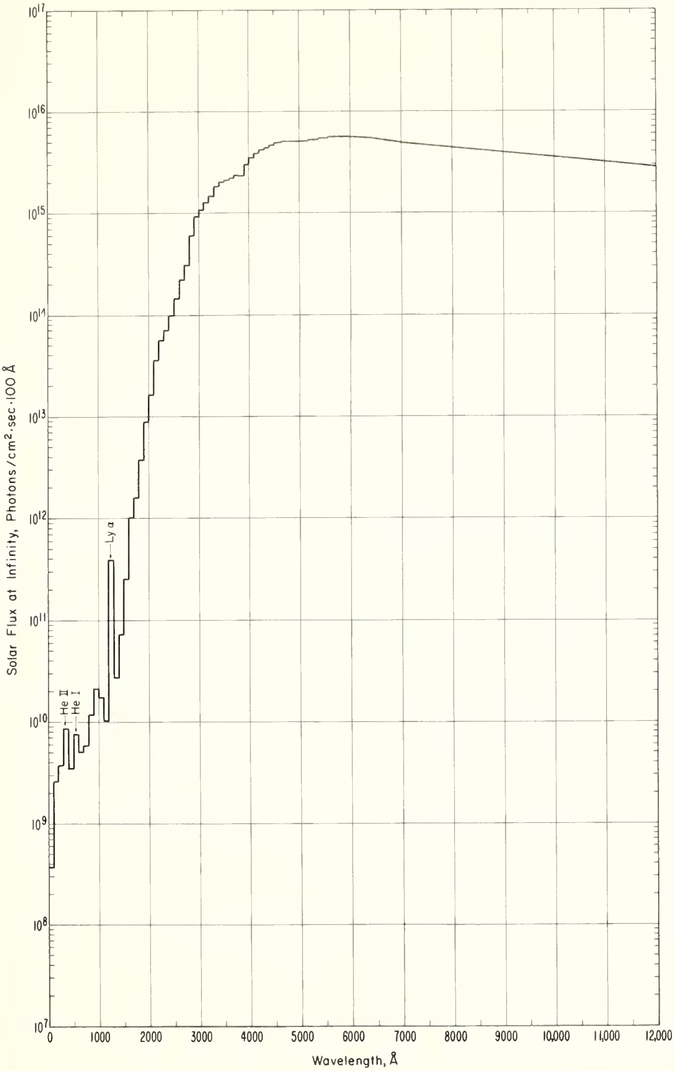

Figure 2. Solar photon flux outside the atmosphere. (These data are also tabulated in table 1.)

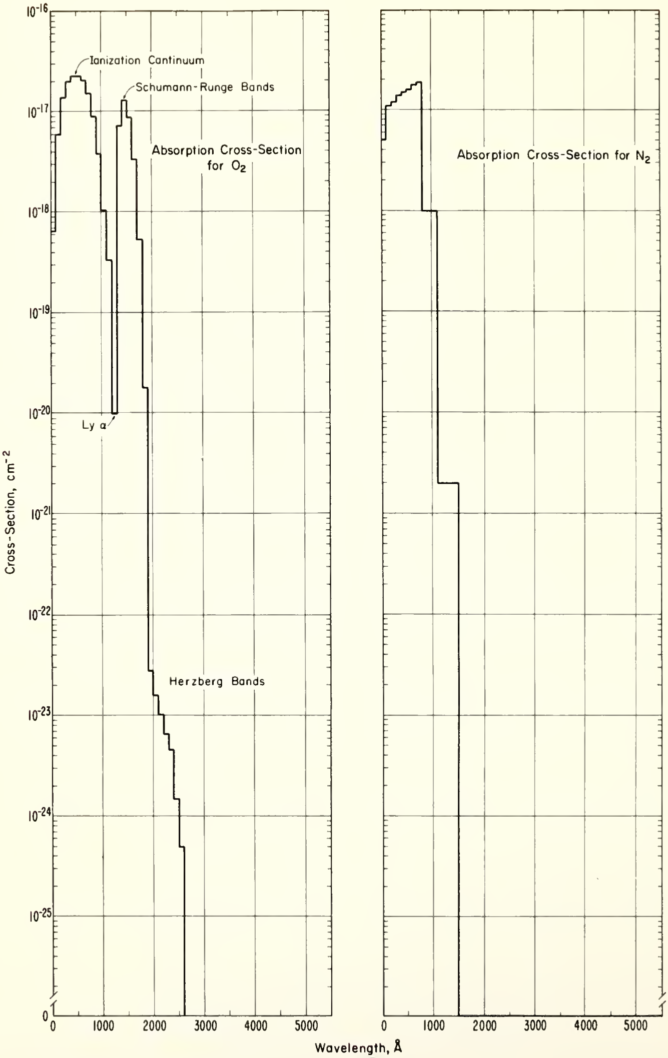

Figure 3. Absorption cross sections for N_2 and O_2. (These data also tabulated in table 1.)

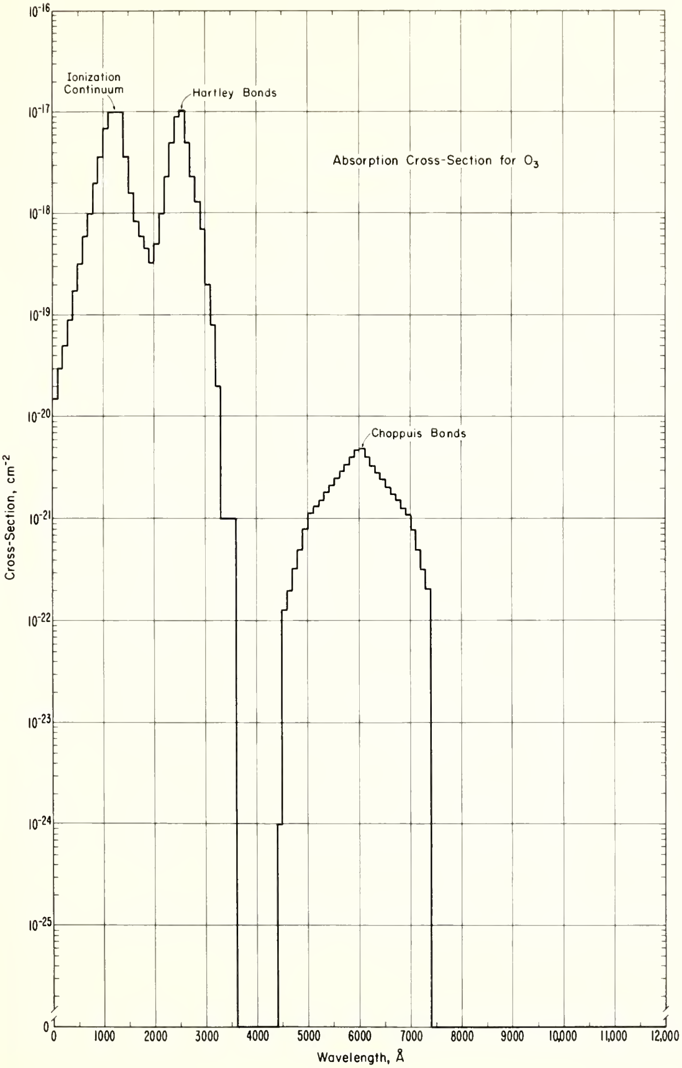

Figure 4. Absorption cross sections for O_3. (These data are also tabulated in table 1.)

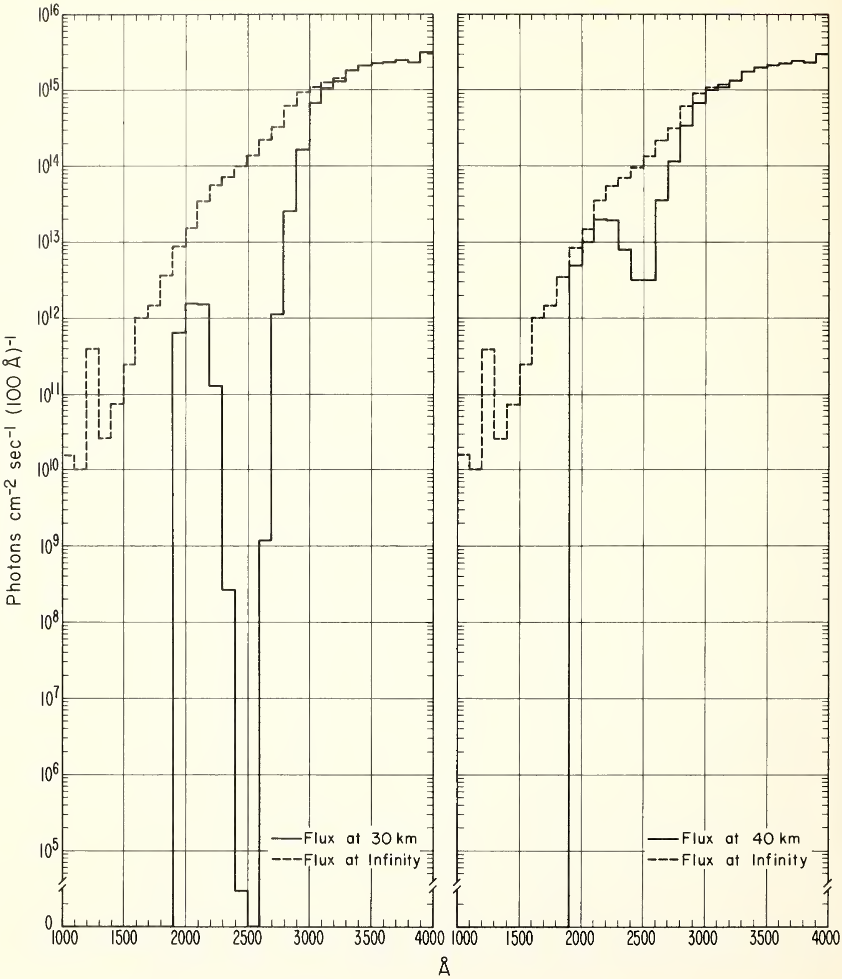

Figure 5. Solar photon flux at 30 and 40 km in the Earth's atmosphere for an overhead sun. Also shown is the flux outside the atmosphere from figure 2.

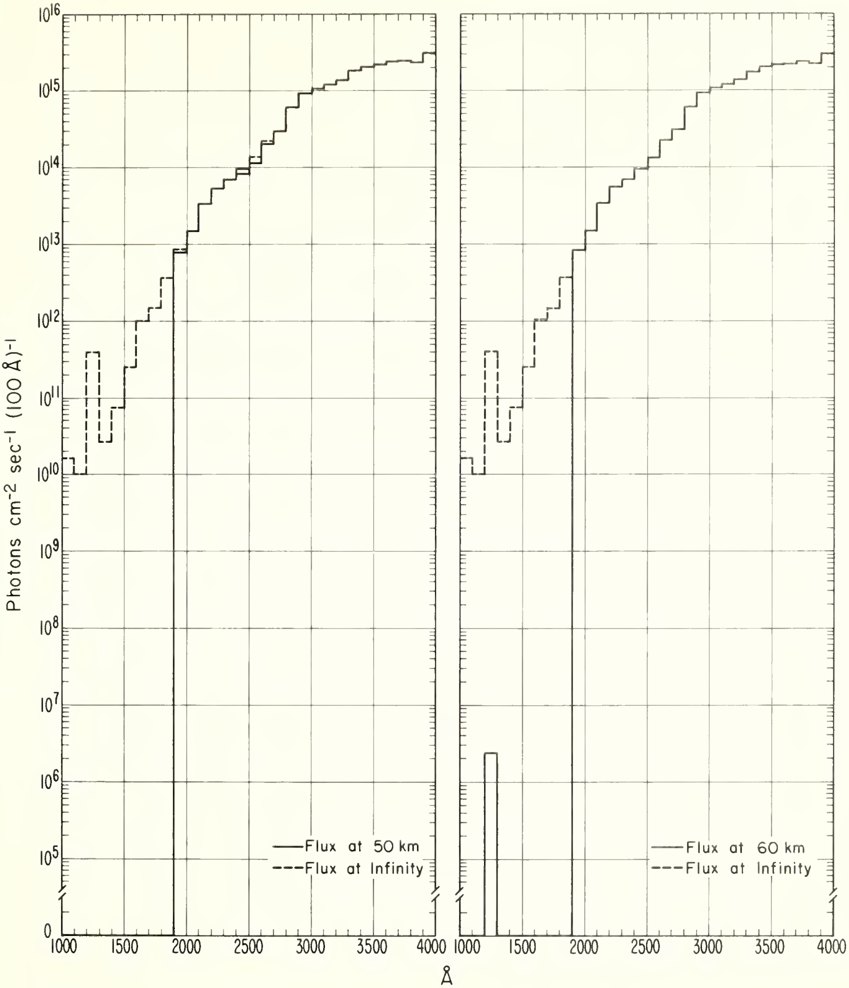

Figure 6. Solar photon flux at 50 and 60 km in the Earth's atmosphere for an overhead sun. Also shown is the flux outside the atmosphere from figure 2.

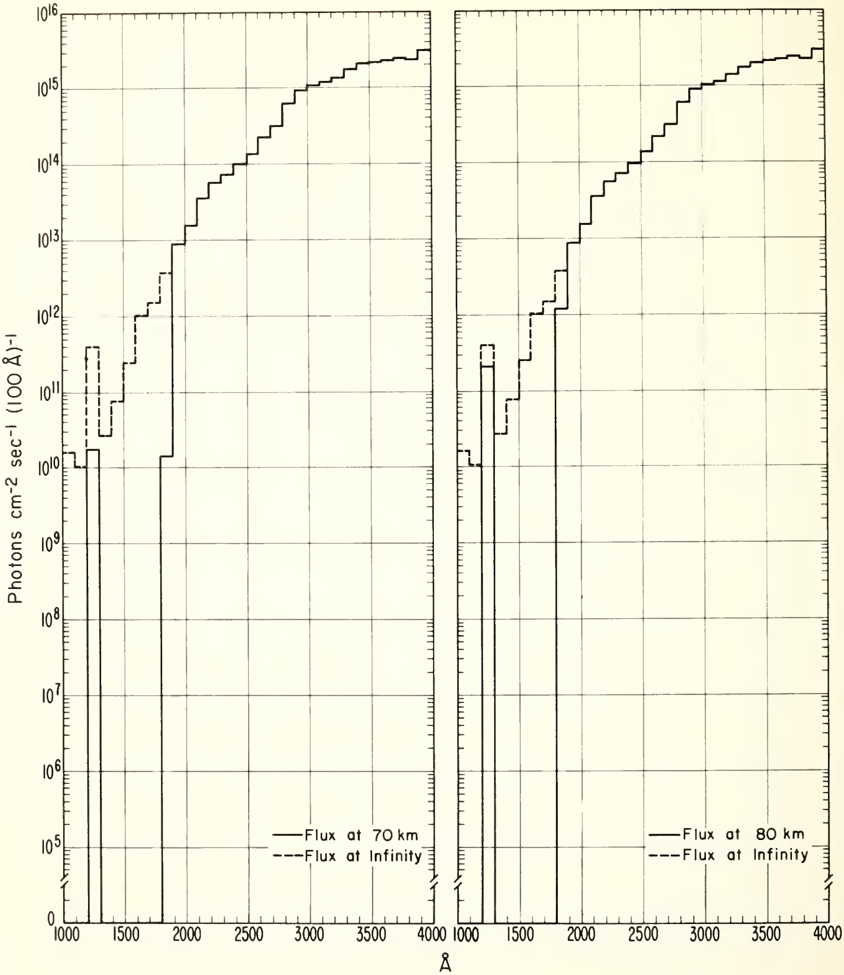

Figure 7. Solar photon flux at 70 and 80 km in the Earth's atmosphere for an overhead sun. Also shown is the flux outside the atmosphere from figure 2.

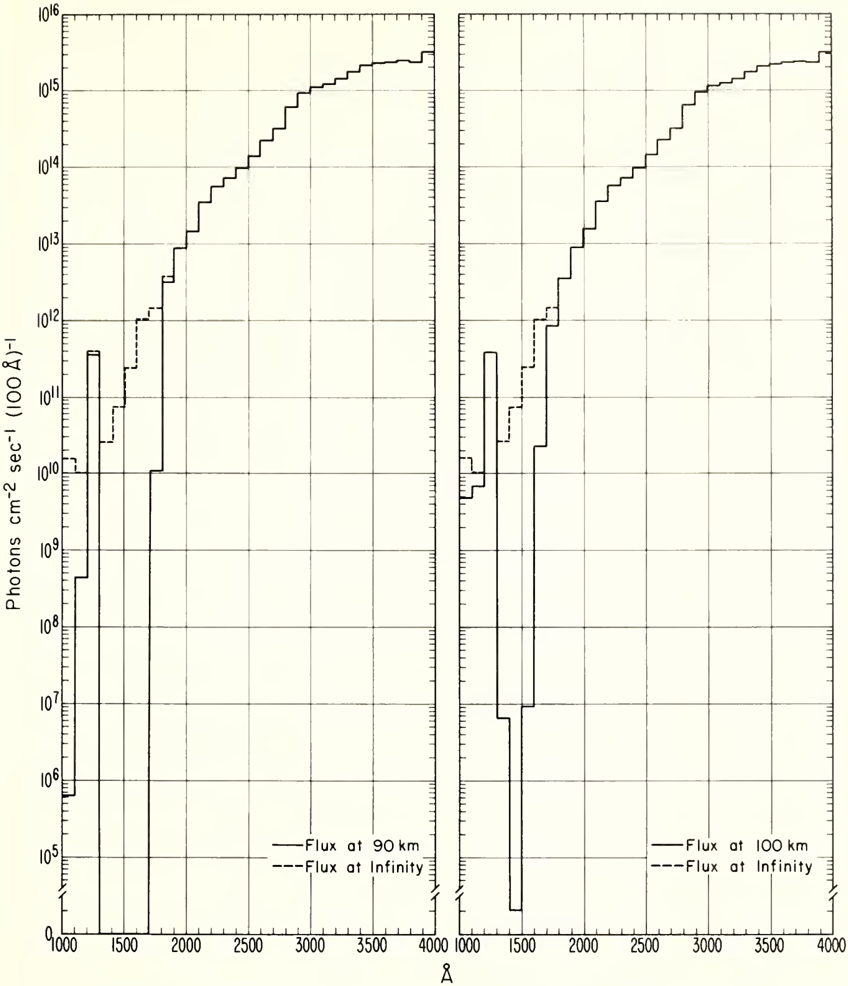

Figure 8. Solar photon flux at 90 and 100 km in the Earth's atmosphere for an overhead sun. Also shown is the flux outside the atmosphere from figure 2.

Table 1. Internal Data*

Wavelength Band(Å)	Solar Photon Flux Outside the Atmosphere (photons/cm2-sec-100Å)	Absorption Coefficients (cm^2/100Å) N_2	O_2	O_3
0-100	3.79(8)	5.0(-18)	6.0(-19)	1.5(-20)
100-200	2.65(9)	1.1(-17)	6.0(-18)	3.0(-20)
200-300	3.80(9)	1.2(-17)	1.4(-17)	5.0(-20)
300-400	8.83(9)	1.4(-17)	2.0(-17)	9.0(-20)
400-500	3.52(9)	1.5(-17)	2.3(-17)	1.7(-19)
500-600	7.71(9)	1.6(-17)	2.3(-17)	3.2(-19)
600-700	5.08(9)	1.8(-17)	2.1(-17)	6.0(-19)
700-800	5.88(9)	1.9(-17)	1.5(-17)	1.0(-18)
800-900	1.20(10)	1.0(-18)	9.0(-18)	2.0(-18)
900-1000	2.16(10)	1.0(-18)	3.8(-18)	3.6(-18)
1000-1100	1.75(10)	1.0(-18)	1.1(-18)	7.0(-18)
1100-1200	1.03(10)	2.0(-21)	3.4(-19)	1.0(-17)
1200-1300	3.94(11)	2.0(-21)	1.0(-20)	1.0(-17)
1300-1400	2.72(10)	2.0(-21)	7.2(-18)	1.0(-17)
1400-1500	7.30(10)	2.0(-21)	1.3(-17)	3.6(-18)
1500-1600	2.54(11)	0.0	8.8(-18)	1.6(-18)
1600-1700	1.03(12)	0.0	3.3(-18)	8.3(-19)
1700-1800	1.59(12)	0.0	5.3(-19)	6.0(-19)
1800-1900	3.74(12)	0.0	1.8(-20)	4.5(-19)
1900-2000	8.88(12)	0.0	2.8(-23)	3.2(-19)
2000-2100	1.64(13)	0.0	1.6(-23)	5.0(-19)
2100-2200	3.58(13)	0.0	1.0(-23)	1.0(-18)
2200-2300	5.68(13)	0.0	6.6(-24)	2.3(-18)
2300-2400	7.10(13)	0.0	4.7(-24)	5.0(-18)
2400-2500	9.90(13)	0.0	1.5(-24)	9.0(-18)
2500-2600	1.45(14)	0.0	5.0(-25)	1.0(-17)
2600-2700	2.21(14)	0.0	0.0	5.0(-18)
2700-2800	3.12(14)	0.0	0.0	2.3(-18)
2800-2900	6.12(14)	0.0	0.0	1.3(-18)
2900-3000	9.38(14)	0.0	0.0	7.0(-19)
3000-3100	1.09(15)	0.0	0.0	2.0(-19)
3100-3200	1.27(15)	0.0	0.0	8.0(-20)
3200-3300	1.49(15)	0.0	0.0	2.0(-20)
3300-3400	1.83(15)	0.0	0.0	1.0(-21)
3400-3500	2.05(15)	0.0	0.0	1.0(-21)
3500-3600	2.12(15)	0.0	0.0	1.0(-21)
3600-3700	2.23(15)	0.0	0.0	0.0
3700-3800	2.40(15)	0.0	0.0	0.0
3800-3900	2.36(15)	0.0	0.0	0.0
3900-4000	3.06(15)	0.0	0.0	0.0

Table 1. Internal Data (Cont'd)

Wavelength Band(Å)	Solar Photon Flux Outside the Atmosphere (photons/cm^2-sec-100Å)	Absorption Coefficients (cm^2/100Å)		
		N_2	O_2	O_3
4000-4100	3.53(15)	0.0	0.0	0.0
4100-4200	3.93(15)	0.0	0.0	0.0
4200-4300	4.25(15)	0.0	0.0	0.0
4300-4400	4.45(15)	0.0	0.0	0.0
4400-4500	4.73(15)	0.0	0.0	1.0(-24)
4500-4600	4.94(15)	0.0	0.0	1.3(-22)
4600-4700	5.00(15)	0.0	0.0	2.0(-22)
4700-4800	5.20(15)	0.0	0.0	3.3(-22)
4800-4900	5.20(15)	0.0	0.0	5.0(-22)
4900-5000	5.20(15)	0.0	0.0	8.0(-22)
5000-5100	5.20(15)	0.0	0.0	1.1(-21)
5100-5200	5.20(15)	0.0	0.0	1.3(-21)
5200-5300	5.30(15)	0.0	0.0	1.5(-21)
5300-5400	5.30(15)	0.0	0.0	1.8(-21)
5400-5500	5.40(15)	0.0	0.0	2.1(-21)
5500-5600	5.40(15)	0.0	0.0	2.5(-21)
5600-5700	5.40(15)	0.0	0.0	2.9(-21)
5700-5800	5.40(15)	0.0	0.0	3.4(-21)
5800-5900	5.40(15)	0.0	0.0	4.0(-21)
5900-6000	5.40(15)	0.0	0.0	4.7(-21)
6000-6100	5.40(15)	0.0	0.0	4.8(-21)
6100-6200	5.40(15)	0.0	0.0	4.0(-21)
6200-6300	5.40(15)	0.0	0.0	3.3(-21)
6300-6400	5.35(15)	0.0	0.0	2.8(-21)
6400-6500	5.35(15)	0.0	0.0	2.4(-21)
6500-6600	5.30(15)	0.0	0.0	2.0(-21)
6600-6700	5.25(15)	0.0	0.0	1.7(-21)
6700-6800	5.20(15)	0.0	0.0	1.5(-21)
6800-6900	5.15(15)	0.0	0.0	1.2(-21)
6900-7000	5.10(15)	0.0	0.0	1.1(-21)
7000-7100	5.05(15)	0.0	0.0	7.8(-22)
7100-7200	5.00(15)	0.0	0.0	5.0(-22)
7200-7300	4.95(15)	0.0	0.0	3.2(-22)
7300-7400	4.90(15)	0.0	0.0	2.0(-22)
7400-7500	4.85(15)	0.0	0.0	0.0
7500-7600	4.80(15)	0.0	0.0	0.0
7600-7700	4.72(15)	0.0	0.0	0.0
7700-7800	4.64(15)	0.0	0.0	0.0
7800-7900	4.56(15)	0.0	0.0	0.0
7900-8000	4.48(15)	0.0	0.0	0.0
8000-8100	4.44(15)	0.0	0.0	0.0
8100-8200	4.39(15)	0.0	0.0	0.0
8200-8300	4.34(15)	0.0	0.0	0.0
8300-8400	4.30(15)	0.0	0.0	0.0

Table 1. Internal Data (Cont'd)

Wavelength Band(Å)	Solar Photon Flux Outside the Atmosphere (photons/cm^2-sec-100Å)	Absorption Coefficients (cm^2/100Å)		
		N_2	O_2	O_3
8400-8500	4.25(15)	0.0	0.0	0.0
8500-8600	4.21(15)	0.0	0.0	0.0
8600-8700	4.17(15)	0.0	0.0	0.0
8700-8800	4.13(15)	0.0	0.0	0.0
8800-8900	4.09(15)	0.0	0.0	0.0
8900-9000	4.05(15)	0.0	0.0	0.0
9000-9100	4.00(15)	0.0	0.0	0.0
9100-9200	3.95(15)	0.0	0.0	0.0
9200-9300	3.90(15)	0.0	0.0	0.0
9300-9400	3.85(15)	0.0	0.0	0.0
9400-9500	3.80(15)	0.0	0.0	0.0
9500-9600	3.77(15)	0.0	0.0	0.0
9600-9700	3.74(15)	0.0	0.0	0.0
9700-9800	3.71(15)	0.0	0.0	0.0
9800-9900	3.68(15)	0.0	0.0	0.0
9900-10000	3.65(15)	0.0	0.0	0.0
10000-10100	3.62(15)	0.0	0.0	0.0
10100-10200	3.58(15)	0.0	0.0	0.0
10200-10300	3.54(15)	0.0	0.0	0.0
10300-10400	3.50(15)	0.0	0.0	0.0
10400-10500	3.46(15)	0.0	0.0	0.0
10500-10600	3.42(15)	0.0	0.0	0.0
10600-10700	3.39(15)	0.0	0.0	0.0
10700-10800	3.36(15)	0.0	0.0	0.0
10800-10900	3.33(15)	0.0	0.0	0.0
10900-11000	3.30(15)	0.0	0.0	0.0
11000-11100	3.26(15)	0.0	0.0	0.0
11100-11200	3.22(15)	0.0	0.0	0.0
11200-11300	3.18(15)	0.0	0.0	0.0
11300-11400	3.14(15)	0.0	0.0	0.0
11400-11500	3.10(15)	0.0	0.0	0.0
11500-11600	3.06(15)	0.0	0.0	0.0
11600-11700	3.01(15)	0.0	0.0	0.0
11700-11800	2.97(15)	0.0	0.0	0.0
11800-11900	2.93(15)	0.0	0.0	0.0
11900-12000	2.88(15)	0.0	0.0	0.0

* after Allen, 1963.

5. REFERENCES

Allen, C. W. (1963), Astrophysical Quantities, 2nd ed., University of London, The Athlone Press.

Hunt, B. G. (1966), Photochemistry of ozone in a moist atmosphere, J. Geophys. Res., 71, 1385-1398.

Kenesha, T. J. (1962), A computer program for solving the reaction rate equations in the E ionospheric region, AFCRL Res. Rpt. 62-828.

Kenesha, T. J. (1963), A solution to the reaction rate equations in the atmosphere below 150 kilometers, AFCRL Res. Rpt. 63-711.

Kenesha, T. J. (1967), A technique for solving the general reaction-rate equations in the atmosphere, AFCRL Res. Rpt. 67-0221.

Romanelli, J. J. (1964), Runge-Kutta methods for the solution of ordinary differential equations, in Mathematical Methods for Digital Computers, A Ralston and H. S. Wilf, editors, John Wiley and Sons, New York.

Swider, W., Jr. (1964), The determination of the optical depth at large solar zenith distances, Planet. Space Sci., 12, 761-782.

```
1
  THIS IS HUNTS REACTION SET, CHANGED TO EXTEND THE SOLAR FLUX DOWN THROUGH
THE UV, WITH THE DEACTIVATION OF O1D INCREASED TO 1.0E-10, AND EQUIL 1.E-8
RUN FIRST TO STEADY STATE, AND THEN RESTART WITH JOSHUA AT 0 AT LAT 90.0 DEG
06 03 05   04 05 01          8.00E-35   3.00E+02   +0.00E+00   +4.45E+02
06 04 01   03 03 01          5.60E-11   3.00E+02   +0.00E+00   -2.85E+03
06 06 05   03 05 01          2.70E-33   3.00E+02   +0.00E+00   +0.00E+00
07 04 01   03 03 01          1.00E-11   3.00E+02   +0.00E+00   +0.00E+00
07 05 01   06 05 01          1.00E-10   3.00E+02   +0.00E+00   +0.00E+00
08 04 01   09 03 01          2.60E-11   3.00E+02   +0.00E+00   +0.00E+00
09 06 01   08 03 01          5.00E-11   3.00E+02   +0.00E+00   +0.00E+00
09 04 01   10 03 01          5.00E-13   3.00E+02   +0.00E+00   +0.00E+00
09 09 01   11 06 01          2.80E-12   3.00E+02   +0.00E+00   +0.00E+00
09 10 01   11 03 01          1.00E-11   3.00E+02   +0.00E+00   +0.00E+00
08 10 01   12 03 01          2.00E-13   3.00E+02   +0.00E+00   +0.00E+00
08 10 01   09 09 01          1.00E-11   3.00E+02   +0.00E+00   +0.00E+00
06 10 01   09 03 01          1.00E-11   3.00E+02   +0.00E+00   +0.00E+00
08 03 05   10 05 01          7.40E-32   3.00E+02   +0.00E+00   +0.00E+00
08 09 05   11 05 01          2.50E-31   3.00E+02   +0.00E+00   +0.00E+00
08 08 05   12 05 01          2.60E-32   3.00E+02   +0.00E+00   +0.00E+00
10 10 01   13 03 01          3.00E-12   3.00E+02   +0.00E+00   +0.00E+00
09 13 01   11 10 01          4.00E-13   3.00E+02   +0.00E+00   +0.00E+00
06 13 01   09 10 01          1.00E-15   3.00E+02   +0.00E+00   +0.00E+00
08 13 01   12 10 01          1.00E-13   3.00E+02   +0.00E+00   +0.00E+00
07 12 01   09 08 01          1.00E-11   3.00E+02   +0.00E+00   +0.00E+00
07 11 01   09 09 01          1.00E-11   3.00E+02   +0.00E+00   +0.00E+00
06 09 05   10 05 01          1.40E-31   3.00E+02   +0.00E+00   +0.00E+00
10 04 01   09 03 03          1.00E-14   3.00E+02   +0.00E+00   +0.00E+00
99
03 01 01   06 06 01
018  025
2.65E-19   1.80E-20   2.80E-23   1.60E-23   1.00E-23   6.60E-24   4.60E-24
3.75E-25
03 01 01   06 07 01
012  018
3.40E-19   1.00E-20   7.20E-18   1.30E-17   8.80E-18   3.30E-18   2.65E-19
03 01 01   07 07 01
001  011
6.00E-19   6.00E-18   1.40E-17   2.00E-17   2.30E-17   2.30E-17   2.10E-17
1.50E-17   9.00E-18   3.80E-18   1.10E-18
04 01 01   03 06 01
032  074
8.00E-20   2.00E-20   1.00E-21   1.00E-21   1.00E-21   0.00E+00   0.00E+00
0.00E+00   0.00E+00   0.00E+00   0.00E+00   0.00E+00   0.00E+00   1.00E-24
1.30E-22   2.00E-22   3.30E-22   5.00E-22   8.00E-22   1.13E-21   1.30E-21
1.50E-21   1.80E-21   2.10E-21   2.50E-21   2.90E-21   3.35E-21   4.00E-21
4.65E-21   4.80E-21   4.00E-21   3.25E-21   2.80E-21   2.40E-21   2.00E-21
1.70E-21   1.50E-21   1.25E-21   1.08E-21   7.80E-22   5.00E-22   3.20E-22
2.05E-22
04 01 01   07 03 01
001  031
1.50E-20   3.00E-20   5.00E-20   9.00E-20   1.70E-19   3.20E-19   6.00E-19
1.00E-18   2.00E-18   3.60E-18   7.00E-18   1.00E-17   1.00E-17   1.00E-17
3.60E-18   1.60E-18   8.30E-19   6.00E-19   4.50E-19   3.20E-19   5.00E-19
1.00E-18   2.30E-18   5.00E-18   9.00E-18   1.00E-17   5.00E-18   2.30E-18
1.30E-18   7.00E-19   2.00E-19
11 01 01   09 08 01
002  019
1.00E-18   1.00E-17   2.00E-17   1.30E-17   1.50E-17   1.80E-17   1.20E-17
1.00E-17   1.00E-17   8.00E-18   5.00E-18   1.40E-17   3.00E-18   7.00E-19
2.00E-18   4.00E-18   2.00E-18   2.00E-19
13 01 01   09 09 01
```

```
002  030
1.00E-18   1.00E-17   2.00E-17   1.30E-17   1.50E-17   1.80E-17   1.20E-17
1.00E-17   1.00E-17   8.00E-18   5.00E-18   1.40E-17   3.00E-18   7.00E-19
2.00E-18   4.00E-18   2.00E-18   1.40E-18   7.00E-19   4.70E-19   3.50E-19
2.60E-19   1.80E-19   1.50E-19   1.00E-19   5.00E-20   3.00E-20   2.00E-20
1.60E-20
99
00  1.000E+02  3.000E+01  1.000E+01
2.12E+02   1.81E+02   1.85E+02   2.17E+02   2.58E+02   2.70E+02   2.48E+02
2.28E+02
03   0 2
04   0 3
05   0 0
06   0 1
07   0 1
08   1 1
09   1 2
10   1 3
11   2 3
12   2 2
13   2 4
0.00E+00   1.80E+03   1.30E+06   2.59E+05   3.30E+07   1  0  1.00E-08  0
+9.00E+01  +0.00E+00  09  22  12  00  00
 3
3.64+012   1.32+013   8.07+013   3.68+014   1.27+015   4.26+015   1.69+016
7.66+016   0.00+000   0.00+000   0.00+000   0.00+000   0.00+000   0.00+000
 4
8.21+006   3.57+007   1.74+008   5.04+008   1.26+010   1.17+011   1.41+012
2.89+012   0.00+000   0.00+000   0.00+000   0.00+000   0.00+000   0.00+000
 5
9.98+012   6.48+013   4.03+014   1.84+015   6.33+015   2.13+016   8.46+016
3.83+017   0.00+000   0.00+000   0.00+000   0.00+000   0.00+000   0.00+000
 6
3.34+012   8.69+011   5.58+010   8.47+009   2.22+010   1.10+010   1.62+009
7.50+007   0.00+000   0.00+000   0.00+000   0.00+000   0.00+000   0.00+000
 7
1.11+002   4.12+001   2.92+001   1.71+001   1.16+002   1.70+002   6.37+001
4.35+000   0.00+000   0.00+000   0.00+000   0.00+000   0.00+000   0.00+000
 8
9.95+007   4.75+008   9.07+007   7.92+006   1.58+006   1.83+005   1.07+003
1.15+000   0.00+000   0.00+000   0.00+000   0.00+000   0.00+000   0.00+000
 9
1.29+002   1.09+004   2.26+005   1.19+006   1.32+006   3.24+006   1.89+006
6.88+005   0.00+000   0.00+000   0.00+000   0.00+000   0.00+000   0.00+000
10
8.00+000   3.49+003   3.92+005   4.71+006   4.29+006   1.27+007   4.72+007
3.35+007   0.00+000   0.00+000   0.00+000   0.00+000   0.00+000   0.00+000
11
1.50-002   1.71+002   7.35+005   6.67+009   3.07+010   1.52+011   8.05+011
1.51+012   0.00+000   0.00+000   0.00+000   0.00+000   0.00+000   0.00+000
12
4.30+005   1.99+008   3.11+009   4.45+009   4.88+008   1.88+008   5.26+007
8.72+005   0.00+000   0.00+000   0.00+000   0.00+000   0.00+000   0.00+000
13
5.50-008   3.56-002   2.41+003   4.94+005   3.87+005   5.67+006   5.97+008
6.91+009   0.00+000   0.00+000   0.00+000   0.00+000   0.00+000   0.00+000
99
''
```

APPENDIX B. SIXBIT OUTPUT FOR THE INPUT SHOWN IN APPENDIX A.

```
 THIS IS HUNTS REACTION SET, CHANGED TO EXTEND THE SOLAR FLUX DOWN THROUGH

THE UV, WITH THE DEACTIVATION OF O1D INCREASED TO 1.0E-10, AND EQUIL 1.E-8

RUN FIRST TO STEADY STATE, AND THEN RESTART WITH JOSHUA AT 0 AT LAT 85.0 DEG
THERE HAVE BEEN 24 REACTIONS INPUT INVOLVING 11 SPECIES.
THE REACTIONS AND THEIR RATE COEFFICIENTS ARE

 1     6+ 3+ 5--- 4+ 5+ 1     K = 8.00-035*(TEMP/3.00+002)** 0.00*EXPF( 4.45+002/TEMP)

 2     6+ 4+ 1--- 3+ 3+ 1     K = 5.60-011*(TEMP/3.00+002)** 0.00*EXPF(-2.85+003/TEMP)

 3     6+ 6+ 5--- 3+ 5+ 1     K = 2.70-033*(TEMP/3.00+002)** 0.00*EXPF( 0.00+000/TEMP)

 4     7+ 4+ 1--- 3+ 3+ 1     K = 1.00-011*(TEMP/3.00+002)** 0.00*EXPF( 0.00+000/TEMP)

 5     7+ 5+ 1--- 6+ 5+ 1     K = 1.00-010*(TEMP/3.00+002)** 0.00*EXPF( 0.00+000/TEMP)

 6     8+ 4+ 1--- 9+ 3+ 1     K = 2.60-011*(TEMP/3.00+002)** 0.00*EXPF( 0.00+000/TEMP)

 7     9+ 6+ 1--- 8+ 3+ 1     K = 5.00-011*(TEMP/3.00+002)** 0.00*EXPF( 0.00+000/TEMP)

 8     9+ 4+ 1---10+ 3+ 1     K = 5.00-013*(TEMP/3.00+002)** 0.00*EXPF( 0.00+000/TEMP)

 9     9+ 9+ 1---11+ 6+ 1     K = 2.80-012*(TEMP/3.00+002)** 0.00*EXPF( 0.00+000/TEMP)

10     9+10+ 1---11+ 3+ 1     K = 1.00-011*(TEMP/3.00+002)** 0.00*EXPF( 0.00+000/TEMP)

11     8+10+ 1---12+ 3+ 1     K = 2.00-013*(TEMP/3.00+002)** 0.00*EXPF( 0.00+000/TEMP)

12     8+10+ 1--- 9+ 9+ 1     K = 1.00-011*(TEMP/3.00+002)** 0.00*EXPF( 0.00+000/TEMP)

13     6+10+ 1--- 9+ 3+ 1     K = 1.00-011*(TEMP/3.00+002)** 0.00*EXPF( 0.00+000/TEMP)
```

```
14     8+ 3+ 5---10+ 5+ 1     K = 7.40-032*(TEMP/3.00+002)** 0.00*EXPF( 0.00+000/TEMP)

15     8+ 9+ 5---11+ 5+ 1     K = 2.50-031*(TEMP/3.00+002)** 0.00*EXPF( 0.00+000/TEMP)

16     8+ 8+ 5---12+ 5+ 1     K = 2.60-032*(TEMP/3.00+002)** 0.00*EXPF( 0.00+000/TEMP)

17    10+10+ 1---13+ 3+ 1     K = 3.00-012*(TEMP/3.00+002)** 0.00*EXPF( 0.00+000/TEMP)

18     9+13+ 1---11+10+ 1     K = 4.00-013*(TEMP/3.00+002)** 0.00*EXPF( 0.00+000/TEMP)

19     6+13+ 1--- 9+10+ 1     K = 1.00-015*(TEMP/3.00+002)** 0.00*EXPF( 0.00+000/TEMP)

20     8+13+ 1---12+10+ 1     K = 1.00-013*(TEMP/3.00+002)** 0.00*EXPF( 0.00+000/TEMP)

21     7+12+ 1--- 9+ 8+ 1     K = 1.00-011*(TEMP/3.00+002)** 0.00*EXPF( 0.00+000/TEMP)

22     7+11+ 1--- 9+ 9+ 1     K = 1.00-011*(TEMP/3.00+002)** 0.00*EXPF( 0.00+000/TEMP)

23     6+ 9+ 5---10+ 5+ 1     K = 1.40-031*(TEMP/3.00+002)** 0.00*EXPF( 0.00+000/TEMP)

24    10+ 4+ 1--- 9+ 3+ 3     K = 1.00-014*(TEMP/3.00+002)** 0.00*EXPF( 0.00+000/TEMP)


THERE HAVE BEEN  7 PHOTOREACTIONS INPUT.
25       3+ PHOTON --- 6+ 6+ 1

26       3+ PHOTON --- 6+ 7+ 1

27       3+ PHOTON --- 7+ 7+ 1
```

```
28     4+ PHOTON --- 3+ 6+ 1

29     4+ PHOTON --- 7+ 3+ 1

30    11+ PHOTON --- 9+ 8+ 1

31    13+ PHOTON --- 9+ 9+ 1
```

----------STARTING VALUES----------
REACTION TERM BLOCK CONTAINS RATE COEFFICIENTS

LOCAL TIME (HOURS,MINUTES,SECONDS) = 12/00/00 ON DAY NUMBER 22, MONTH NUMBER 6
COMPUTER TIME= 135.641 SECONDS
RUNNING TIME = 0.000+000 SECONDS
ZENITH ANGLE = 65.27 DEGREES, LATITUDE = 85.00LONGITUDE = 0.00

REACTION TERMS--------

HEIGHT	STEP SIZE TERM(8) TERM(16) TERM(24)	TERM(1) TERM(9) TERM(17) TERM(25)	TERM(2) TERM(10) TERM(18) TERM(26)	TERM(3) TERM(11) TERM(19) TERM(27)	TERM(4) TERM(12) TERM(20) TERM(28)	TERM(5) TERM(13) TERM(21) TERM(29)	TERM(6) TERM(14) TERM(22) TERM(30)	TERM(7) TERM(15) TERM(23) TERM(31)
100.0	1.0000+000	6.527-034	8.124-017	2.700-033	1.000-011	1.000-010	2.600-011	5.000-011
	5.000-013	2.800-012	1.000-011	2.000-013	1.000-011	1.000-011	7.400-032	2.500-031
	2.600-032	3.000-012	4.000-013	1.000-015	1.000-013	1.000-011	1.000-011	1.400-031
	1.000-014	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000
90.0	1.0000+000	9.350-034	8.125-018	2.700-033	1.000-011	1.000-010	2.600-011	5.000-011
	5.000-013	2.800-012	1.000-011	2.000-013	1.000-011	1.000-011	7.400-032	2.500-031
	2.600-032	3.000-012	4.000-013	1.000-015	1.000-013	1.000-011	1.000-011	1.400-031
	1.000-014	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000
80.0	1.0000+000	8.866-034	1.142-017	2.700-033	1.000-011	1.000-010	2.600-011	5.000-011
	5.000-013	2.800-012	1.000-011	2.000-013	1.000-011	1.000-011	7.400-032	2.500-031
	2.600-032	3.000-012	4.000-013	1.000-015	1.000-013	1.000-011	1.000-011	1.400-031
	1.000-014	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000
70.0	1.0000+000	6.219-034	1.107-016	2.700-033	1.000-011	1.000-010	2.600-011	5.000-011
	5.000-013	2.800-012	1.000-011	2.000-013	1.000-011	1.000-011	7.400-032	2.500-031
	2.600-032	3.000-012	4.000-013	1.000-015	1.000-013	1.000-011	1.000-011	1.400-031
	1.000-014	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000
60.0	1.0000+000	4.489-034	8.928-016	2.700-033	1.000-011	1.000-010	2.600-011	5.000-011
	5.000-013	2.800-012	1.000-011	2.000-013	1.000-011	1.000-011	7.400-032	2.500-031
	2.600-032	3.000-012	4.000-013	1.000-015	1.000-013	1.000-011	1.000-011	1.400-031
	1.000-014	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000
50.0	1.0000+000	4.158-034	1.459-015	2.700-033	1.000-011	1.000-010	2.600-011	5.000-011
	5.000-013	2.800-012	1.000-011	2.000-013	1.000-011	1.000-011	7.400-032	2.500-031
	2.600-032	3.000-012	4.000-013	1.000-015	1.000-013	1.000-011	1.000-011	1.400-031
	1.000-014	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000
40.0	1.0000+000	4.812-034	5.719-016	2.700-033	1.000-011	1.000-010	2.600-011	5.000-011
	5.000-013	2.800-012	1.000-011	2.000-013	1.000-011	1.000-011	7.400-032	2.500-031
	2.600-032	3.000-012	4.000-013	1.000-015	1.000-013	1.000-011	1.000-011	1.400-031
	1.000-014	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000
30.0	1.0000+000	5.633-034	2.087-016	2.700-033	1.000-011	1.000-010	2.600-011	5.000-011
	5.000-013	2.800-012	1.000-011	2.000-013	1.000-011	1.000-011	7.400-032	2.500-031
	2.600-032	3.000-012	4.000-013	1.000-015	1.000-013	1.000-011	1.000-011	1.400-031
	1.000-014	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000	0.000+000

SPECIES DENSITIES----

HEIGHT	Y(2)	Y(3)	Y(4)	Y(5)	Y(6)	Y(7)	Y(8)	Y(9)
	Y(10)	Y(11)	Y(12)	Y(13)	Y(14)	Y(15)	Y(16)	Y(17)
100.0	1.00000000	3.640+012	8.210+006	9.980+012	3.340+012	1.110+002	9.950+007	1.290+002
	8.000+000	1.500-002	4.300+005	5.500-008	0.000+000	0.000+000	0.000+000	0.000+000
90.0	1.00000000	1.320+013	3.570+007	6.480+013	8.690+011	4.120+001	4.750+008	1.090+004
	3.490+003	1.710+002	1.990+008	3.560-002	0.000+000	0.000+000	0.000+000	0.000+000
80.0	1.00000000	8.070+013	1.740+008	4.030+014	5.580+010	2.920+001	9.070+007	2.260+005
	3.920+005	7.350+005	3.110+009	2.410+003	0.000+000	0.000+000	0.000+000	0.000+000
70.0	1.00000000	3.680+014	5.040+008	1.840+015	8.470+009	1.710+001	7.920+006	1.190+006
	4.710+006	6.670+009	4.450+009	4.940+005	0.000+000	0.000+000	0.000+000	0.000+000
60.0	1.00000000	1.270+015	1.260+010	6.330+015	2.220+010	1.160+002	1.580+006	1.320+006
	4.290+006	3.070+010	4.880+008	3.870+005	0.000+000	0.000+000	0.000+000	0.000+000
50.0	1.00000000	4.260+015	1.170+011	2.130+016	1.100+010	1.700+002	1.830+005	3.240+006
	1.270+007	1.520+011	1.880+008	5.670+006	0.000+000	0.000+000	0.000+000	0.000+000
40.0	1.00000000	1.690+016	1.410+012	8.460+016	1.620+009	6.370+001	1.070+003	1.890+006
	4.720+007	8.050+011	5.260+007	5.970+008	0.000+000	0.000+000	0.000+000	0.000+000
30.0	1.00000000	7.660+016	2.890+012	3.830+017	7.500+007	4.350+000	1.150+000	6.880+005
	3.350+007	1.510+012	8.720+005	6.910+009	0.000+000	0.000+000	0.000+000	0.000+000

LOCAL TIME (HOURS,MINUTES,SECONDS) = 12/30/00 ON DAY NUMBER 22, MONTH NUMBER 6
COMPUTER TIME= 138.180 SECONDS
RUNNING TIME = 1.800+003 SECONDS
ZENITH ANGLE = 65.27 DEGREES, LATITUDE = 85.00LONGITUDE = 0.00

REACTION TERMS--------

HEIGHT	STEP SIZE / TERM(8) / TERM(16) / TERM(24)	TERM(1) / TERM(9) / TERM(17) / TERM(25)	TERM(2) / TERM(10) / TERM(18) / TERM(26)	TERM(3) / TERM(11) / TERM(19) / TERM(27)	TERM(4) / TERM(12) / TERM(20) / TERM(28)	TERM(5) / TERM(13) / TERM(21) / TERM(29)	TERM(6) / TERM(14) / TERM(22) / TERM(30)	TERM(7) / TERM(15) / TERM(23) / TERM(31)
100.0	3.0000+002	7.919+004	2.226+003	3.006+005	9.077-003	1.104+005	2.122+004	2.150+004
	5.280-004	4.639-008	1.031-008	1.594-004	7.969-003	2.675+002	2.675+002	3.196-008
	2.569-003	1.924-010	2.838-018	1.841-010	5.484-013	4.758-004	1.660-011	6.007-004
	6.569-007	2.663+005	5.803+004	5.733+001	3.479+003	5.226+004	8.871-008	7.764-012
90.0	1.2000+002	6.950+005	2.563+002	1.321+005	1.518-002	2.710+005	4.482+005	4.783+005
	1.998-001	3.393-004	3.805-004	3.283-001	1.641+001	3.004+004	3.006+004	8.467-005
	3.799-001	3.585-005	1.508-010	2.976-005	1.626-006	8.321-002	7.151-008	8.678-002
	1.255-003	5.703+005	3.979+004	3.227-004	1.540+004	2.312+005	8.005-004	4.739-006
80.0	1.2000+002	1.611+006	1.125+002	3.396+003	5.246-002	1.199+006	4.153+005	6.337+005
	1.999+001	1.441-001	8.842-001	7.064+000	3.532+002	2.177+005	2.181+005	2.071-003
	8.603-002	4.558-001	2.160-004	1.330-001	2.157-002	9.255-001	2.187-004	7.150-001
	6.871-001	5.765+005	7.915+004	1.286-032	7.476+004	1.120+006	1.054+000	3.156-001
70.0	1.2000+002	3.647+006	5.011+002	3.727+002	9.360-002	3.296+006	1.091+005	5.121+005
	3.089+002	3.915+000	5.490+001	7.455+000	3.728+002	4.021+005	4.023+005	4.367-003
	3.083-003	6.468+001	2.313-001	4.236+000	3.927-001	7.972-001	1.195+000	2.638+000
	2.426+001	5.857+005	1.713+003	3.638-171	2.216+005	3.305+006	4.350+001	6.361+001
60.0	1.2000+002	8.073+007	2.550+005	8.553+003	1.499+001	7.432+007	5.277+005	1.474+006
	8.415+003	4.865+000	5.621+001	1.356+000	6.779+001	9.539+005	9.457+005	3.316-003
	4.159-004	5.455+001	2.031-001	8.617+000	6.123-002	5.729-001	3.604+001	2.613+001
	5.445+002	1.912+006	1.793-005	0.000+000	5.413+006	7.442+007	6.069-007	4.683+001
50.0	3.0000+002	4.158+008	1.888+006	6.986+003	2.000+002	3.627+008	5.571+005	1.784+006
	1.900+005	2.933+001	4.110+002	4.634-001	2.317+001	1.400+006	1.225+006	3.144-003
	1.843-005	4.839+002	7.323+000	6.235+001	1.032-001	3.202-001	2.589+002	1.064+002
	1.491+004	4.444+006	3.891-034	0.000+000	4.962+007	3.630+008	1.944-035	4.223+002
40.0	3.6000+002	1.114+009	1.305+006	5.985+002	8.968+002	5.383+008	3.919+004	1.523+005
	1.326+006	9.909+000	8.864+002	1.008-002	5.039-001	7.627+005	1.132+005	4.255-005
	2.516-009	6.660+003	4.492+002	9.662+002	6.384-002	3.347-002	5.122+002	3.607+001
	6.641+005	2.424+006	9.395-151	0.000+000	5.727+008	5.383+008	6.265-152	5.333+003
30.0	6.0000+002	1.237+009	4.515+004	5.798+000	1.258+002	1.667+008	8.589+001	2.563+003
	9.889+005	1.312+000	2.284+002	7.629-006	3.814-004	2.498+004	2.482+003	7.494-008
	1.302-014	3.339+003	1.892+003	5.174+002	7.900-004	3.796-005	6.573+001	2.748+000
	9.639+005	9.573+004	0.000+000	0.000+000	1.070+009	1.667+008	0.000+000	9.838+002

SPECIES DENSITIES----

HEIGHT	Y(2) / Y(10)	Y(3) / Y(11)	Y(4) / Y(12)	Y(5) / Y(13)	Y(6) / Y(14)	Y(7) / Y(15)	Y(8) / Y(16)	Y(9) / Y(17)
100.0	0.99991427	3.640+012	8.203+006	9.980+012	3.340+012	1.106+002	9.950+007	1.287+002
	8.008+000	1.500-002	4.300+005	5.511-008	0.000+000	0.000+000	0.000+000	0.000+000
90.0	1.00016882	1.320+013	3.630+007	6.480+013	8.690+011	4.181+001	4.749+008	1.101+004
	3.457+003	1.710+002	1.990+008	3.423-002	0.000+000	0.000+000	0.000+000	0.000+000
80.0	1.00001062	8.070+013	1.763+008	4.030+014	5.587+010	2.977+001	9.061+007	2.269+005
	3.897+005	7.350+005	3.110+009	2.380+003	0.000+000	0.000+000	0.000+000	0.000+000
70.0	0.99999515	3.680+014	5.250+008	1.840+015	8.704+009	1.799+001	8.049+006	1.181+006
	4.632+006	6.670+009	4.450+009	4.878+005	0.000+000	0.000+000	0.000+000	0.000+000
60.0	0.99999900	1.270+015	1.277+010	6.330+015	2.238+010	1.176+002	1.590+006	1.318+006
	4.262+006	3.070+010	4.880+008	3.851+005	0.000+000	0.000+000	0.000+000	0.000+000
50.0	0.99999815	4.260+015	1.175+011	2.130+016	1.103+010	1.705+002	1.826+005	3.234+006
	1.269+007	1.520+011	1.880+008	5.656+006	0.000+000	0.000+000	0.000+000	0.000+000
40.0	0.99999838	1.690+016	1.409+012	8.460+016	1.618+009	6.362+001	1.069+003	1.880+006
	4.708+007	8.050+011	5.260+007	5.968+008	0.000+000	0.000+000	0.000+000	0.000+000
30.0	0.99999770	7.660+016	2.887+012	3.830+017	7.485+007	4.351+000	1.140+000	6.825+005
	3.326+007	1.510+012	8.720+005	6.910+009	0.000+000	0.000+000	0.000+000	0.000+000

MAJOR DESTRUCTION REACTIONS--------

HEIGHT	NO. 2	NO. 3	NO. 4	NO. 5	NO. 6	NO. 7	NO. 8	NO. 9
	NO. 10	NO. 11	NO. 12	NO. 13	NO. 14	NO. 15	NO. 16	NO. 17
100.0	0/ 0, 0, 0	25/01,26,00	29/06,00,00	-1/-1,-1,-1	03/01,00,00	05/00,00,00	06/00,00,00	07/00,00,00
	13/00,00,00	30/00,00,00	21/00,00,00	19/00,00,00	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0
90.0	0/ 0, 0, 0	01/25,00,00	06/29,00,00	-1/-1,-1,-1	01/07,03,00	05/00,00,00	06/00,00,00	07/00,00,00
	13/00,00,00	30/00,00,00	21/00,00,00	19/31,00,00	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0
80.0	0/ 0, 0, 0	01/25,14,00	29/06,00,00	-1/-1,-1,-1	01/07,13,00	05/00,00,00	06/14,00,00	07/00,00,00
	13/00,00,00	30/00,00,00	21/00,00,00	31/19,00,00	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0
70.0	0/ 0, 0, 0	01/25,14,00	29/00,00,00	-1/-1,-1,-1	01/07,13,00	05/00,00,00	14/06,00,00	07/00,00,00
	13/00,00,00	30/00,00,00	21/00,00,00	31/00,00,00	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0
60.0	0/ 0, 0, 0	01/00,00,00	29/00,00,00	-1/-1,-1,-1	01/00,00,00	05/00,00,00	14/06,00,00	07/00,00,00
	13/00,00,00	22/00,00,00	21/00,00,00	31/19,00,00	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0
50.0	0/ 0, 0, 0	01/00,00,00	29/28,00,00	-1/-1,-1,-1	01/00,00,00	05/00,00,00	14/06,00,00	07/08,00,00
	13/00,00,00	22/00,00,00	21/00,00,00	31/19,00,00	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0
40.0	0/ 0, 0, 0	01/00,00,00	28/29,00,00	-1/-1,-1,-1	01/00,00,00	05/00,00,00	14/06,00,00	08/07,00,00
	13/24,00,00	22/00,00,00	21/00,00,00	31/19,00,00	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0
30.0	0/ 0, 0, 0	01/00,00,00	28/29,00,00	-1/-1,-1,-1	01/00,00,00	05/00,00,00	14/00,00,00	08/00,00,00
	24/00,00,00	22/00,00,00	21/00,00,00	18/31,19,00	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0

MAJOR PRODUCTION REACTIONS-------

HEIGHT	NO. 2 / NO. 10	NO. 3 / NO. 11	NO. 4 / NO. 12	NO. 5 / NO. 13	NO. 6 / NO. 14	NO. 7 / NO. 15	NO. 8 / NO. 16	NO. 9 / NO. 17
100.0	0/ 0, 0, 0	03/29,00,00	01/00,00,00	-1/-1,-1,-1	25/05,26,00	26/29,00,00	07/00,00,00	06/00,00,00
	14/00,00,00	09/15,10,00	16/00,00,00	17/00,00,00	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0
90.0	0/ 0, 0, 0	07/06,29,03	01/00,00,00	-1/-1,-1,-1	25/05,00,00	29/26,00,00	07/00,00,00	06/00,00,00
	14/00,00,00	10/09,15,00	16/11,00,00	17/00,00,00	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0
80.0	0/ 0, 0, 0	29/07,06,13	01/00,00,00	-1/-1,-1,-1	05/25,00,00	29/00,00,00	07/00,00,00	06/13,00,00
	14/00,00,00	10/09,00,00	11/00,00,00	17/00,00,00	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0
70.0	0/ 0, 0, 0	29/07,13,00	01/00,00,00	-1/-1,-1,-1	05/25,00,00	29/00,00,00	07/00,00,00	13/06,00,00
	14/00,00,00	10/00,00,00	11/00,00,00	17/00,00,00	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0
60.0	0/ 0, 0, 0	29/00,00,00	01/00,00,00	-1/-1,-1,-1	05/00,00,00	29/00,00,00	07/00,00,00	13/06,00,00
	14/00,00,00	10/00,00,00	11/00,00,00	17/00,00,00	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0
50.0	0/ 0, 0, 0	29/28,00,00	01/00,00,00	-1/-1,-1,-1	05/28,00,00	29/00,00,00	07/00,00,00	13/06,00,00
	14/08,00,00	10/00,00,00	11/20,00,00	17/00,00,00	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0
40.0	0/ 0, 0, 0	28/29,00,00	01/00,00,00	-1/-1,-1,-1	28/05,00,00	29/00,00,00	07/00,00,00	13/24,00,00
	08/00,00,00	10/18,00,00	20/11,00,00	17/00,00,00	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0
30.0	0/ 0, 0, 0	28/29,00,00	01/00,00,00	-1/-1,-1,-1	28/05,00,00	29/00,00,00	07/00,00,00	24/00,00,00
	08/00,00,00	18/10,00,00	20/00,00,00	17/00,00,00	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0	0/ 0, 0, 0

APPENDIX C. SIXBIT LISTING

```
      PROGRAM SIXBIT                                                    SIXBIT
      COMMON /A/      A(51)           ,ARK(4)         ,B(51)          , SIXBIT
     1 BRK(4)        ,CRK(4)          ,F(20)          ,FF(20)         , SIXBIT
     2 FLUX(120)     ,FLXINF(120)     ,FRATE(20)      ,FTERM(40)      , SIXBIT
     3 FTERMW(40)    ,G(20)           ,GG(20)         ,HBAR(20)       , SIXBIT
     4 H2HOLD(20)    ,H3HOLD(20)      ,IATOM(10,20)   ,IGAIN(4)       , SIXBIT
     5 IGAINA(20,20) ,IGAINB(20,20)   ,IGAINC(20,20)  ,IGAIND(20,20)  , SIXBIT
     6 ILOSS(4)      ,ILOSSA(20,20)   ,ILOSSB(20,20)  ,ILOSSC(20,20)  , SIXBIT
     7 ILOSSD(20,20) ,INFORM(30)      ,JPULL(20)      ,K(1000)        , SIXBIT
     8 K0(51)        ,KABS(20,120)    ,KABSN2(120)    ,KABSO2(120)    , SIXBIT
     9 KABSO3(120)   ,KRK(80)         ,L1(51)         ,L2(51)         , SIXBIT
     X L3(51)        ,LL1(21)         ,LL2(21)        ,LL3(21)        , SIXBIT
     1 N(51)         ,NASRAS(1000)    ,NFTERM(20)     ,NTERM(20)      , SIXBIT
     2 NUSRUS(400)   ,PHOLD(20,20)    ,PROHLD(71,20)  ,QOLD(20)       , SIXBIT
     3 QRK(20)       ,R1(51)          ,R2(51)         ,R3(51)         , SIXBIT
     4 RR1(21)       ,RR2(21)         ,RR3(21)        ,SUMIN(10,20)   , SIXBIT
     5 TABO2(87)     ,TABO3(87)       ,TEMP(20)       ,TERM(100)      , SIXBIT
     6 TERMW(100)    ,Y(20)           ,YHOLD(20,20)   ,YLAST(20,20)   , SIXBIT
     7 YNEW(20)      ,YOLD(20)        ,YPRINT(20,20)  ,YSTART(20,20)  , SIXBIT
     8 Y3SUN(20)     ,Y4SUN(20)       ,Z(20)          ,ZPRINT(20)       SIXBIT
      COMMON /B/  ALPHA , EQUIL , H     , HCOUNT, IFLAG1, IFLAG2,       SIXBIT
     2            IFLAG3, IFLAG4, IFLAG5, IFLAG6, IFLAG7, IFLAG8,       SIXBIT
     3            IFLAG9, IJZX  , ITCHXX, IUNIT , JCNTRL, JFLAG2,       SIXBIT
     4            JFLAG6, JOSHUA, JZEND , JZMAIN, LAT   , LONG  ,       SIXBIT
     5            NFREAC, NREAC , NSPEC , PI    , T     , TDAY  ,       SIXBIT
     6            THOUR , TMAX1 , TMAX2 , TMIN  , TMONTH, TOLD  ,       SIXBIT
     7            TPRINT, TQUIT , TSEC  , ZBOTUM, ZSTEP , ZTOP          SIXBIT
      INTEGER HCOUNT,PHOLD,R1,R2,R3,RR1,RR2,RR3,RX1,RX2,RX3,            SIXBIT
     1 TMONTH,TDAY,THOUR,TMIN,TSEC,TX                                   SIXBIT
      REAL K0,N,KABS,K,KRK,LAT,LONG,K5,K5MAX,KABSO2,KABSO3,KABSN2       SIXBIT
C DATA INPUT FORMATS, IN ORDER                                          SIXBIT
C 1.  JFLAG2 = 0 FOR FRESH START,                                       SIXBIT
C            = 1 FOR RESTART FROM TAPE.                                 SIXBIT
C 2.  JJFLAG = 0 FOR NO MICROFILM,                                      SIXBIT
C            = 1 TO GET MICROFILM PLOTS.                                SIXBIT
C 3.  THREE CARDS OF ALPHANUMERIC INFORMATION TO YOURSELF TO BE         SIXBIT
C     PRINTED OUT AT THE FIRST OF THE OUTPUT.                           SIXBIT
C 4.  NORMAL REACTIONS WITH RATE COEFFICIENTS, READ IN REACTS. EACH     SIXBIT
C     CARD CONTAINS L1,L2,L3,R1,R2,R3,K0,A,N,B, WHERE L1,L2,L3, ARE THE SIXBIT
C     SPECIES NUMBERS ON THE LEFT SIDE OF THE REACTION. R1,R2,R3 ARE    SIXBIT
C     THE SPECIES NUMBERS ON THE RIGHT SIDE, AND THE RATE COEFFICIENT ISSIXBIT
C     GIVEN BY K=K0 * (TEMP/A)**N * EXPF(B/TEMP).                       SIXBIT
C     FORMAT (3(I2,1X),1X,3(I2,1X),6X,2(E8.2,2X),2(E9.2,2X))            SIXBIT
C     SAMPLE ( / IS EDGE OF CARD)                                       SIXBIT
C     /07 03 09  05 08 01       4.50E-08  3.00E+02  -1.50E+00  -9.37E+02SIXBIT
C     THIS SET MUST BE TERMINATED WITH A CARD CONTAINING A 99 IN        SIXBIT
C     COLUMNS 1 AND 2.                                                  SIXBIT
C                                                                       SIXBIT
C 5.  PHOTOREACTIONS WITH CROSS-SECTIONS AS A FUNCTION OF WAVE-LENGTH,  SIXBIT
C     READ IN REACTS.                                                   SIXBIT
C     FIRST CARD - LL1,LL2,LL3,RR1,RR2,RR3 - SPECIES NUMBERS AS ABOVE.  SIXBIT
C     SECOND CARD - LAMDAX,LAMDAY - 1ST AND LAST WAVELENGTH INTERVAL    SIXBIT
C     NUMBERS FOR WHICH REACTION CROSS-SECTIONS ARE GIVEN.              SIXBIT
C     THIRD AND FOLLOWING CARDS - KABS - THE REACTION CROSS-SECTIONS.   SIXBIT
C     FORMAT (3(I2,1X),1X,3(I2,1X))                                     SIXBIT
C     FORMAT (I3,2X,I3)                                                 SIXBIT
C     FORMAT (7(E8.2,2X))                                               SIXBIT
C     SAMPLE                                                            SIXBIT
C     /03 01 01  02 09 01                                               SIXBIT
C     /047  112                                                         SIXBIT
C     /4.93E-23  4.92E-23  4.91E-23  4.90E-23                           SIXBIT
C     THIS SET MUST BE TERMINATED WITH A 99.                            SIXBIT
```

```
C                                                                        SIXBIT
C 6.  ALTITUDE INFORMATION, READ IN HEIGHTS.  THIS CAN BE GIVEN EITHER   SIXBIT
C     AS THE TOP AND BOTTOM ALTITUDES AND THE STEP SIZE (WHICH ALLOWS    SIXBIT
C     ONLY CONSTANT SPACING), OR AS A LIST OF ALTITUDES (WHICH CAN BE    SIXBIT
C     ANY SPACING, BUT MUST BE IN ORDER WITH THE HIGHEST ALTITUDE FIRST)SIXBIT
C     THE FIRST VALUE ON THE CARD IS IFLAG6. IF IFLAG6=0, THEN THE NEXT  SIXBIT
C     THREE VALUES ARE TAKEN TO BE THE TOP AND BOTTOM ALTITIDES AND      SIXBIT
C     THE STEP SIZE, ALL IN KILOMETERS. IF IFLAG6 IS NOT ZERO, IT GIVES  SIXBIT
C     THE NUMBER OF ALTITUDE VALUES (IN KILOMETERS) WHICH FOLLOW.        SIXBIT
C     ZERO OR NEGATIVE VALUES OF ALTITUDES ARE NOT ALLOWED.              SIXBIT
C     FORMAT (I2,6(2X,E9.3)) - FIRST CARD                                SIXBIT
C     FORMAT (6(E9.3,2X)) - ALL OTHER CARDS                              SIXBIT
C     SAMPLE NO. 1 (CONSTANT STEP SIZE)                                  SIXBIT
C     /00  1.000E+02  3.000E+01  1.000E+01                               SIXBIT
C     SAMPLE NO. 2 (VARIABLE STEP SIZE)                                  SIXBIT
C     /04  1.000E+02  9.500E+01  9.000E+01  8.000E+01                    SIXBIT
C                                                                        SIXBIT
C 7.  TEMPERATURE PROFILE, READ IN ALGEBRA.                              SIXBIT
C     FORMAT (7(E8.2,2X))                                                SIXBIT
C     SAMPLE                                                             SIXBIT
C     /1.23E+02  1.37E+02  1.46E+02  1.70E+02                            SIXBIT
C                                                                        SIXBIT
C 8.  THE NUMBER OF ATOMS OF EACH DIFFERENT KIND PER MOLECULE. ON EACH   SIXBIT
C     CARD IS THE SPECIES NUMBER, THE NUMBER OF ATOMS OF THE FIRST       SIXBIT
C     KIND PER MOLECULE, THE NUMBER OF ATOMS OF THE SECOND KIND PER      SIXBIT
C     MOLECULE, ETC., OUT TO A MAXIMUM OF TEN KINDS OF ATOMS. THE        SIXBIT
C     KINDS OF ATOMS CAN BE ANYTHING, SUCH AS H ATOMS, NUMBER OF         SIXBIT
C     UNBALANCED ELECTRONS, ETC., BUT SHOULD BE ORDERED WITH LEAST       SIXBIT
C     NUMEROUS KIND FIRST OUT TO MOST NUMEROUS KIND (WHICH WILL NORMALLYSIXBIT
C     BE THE TOTAL NUMBER OF ATOMS) LAST.                                SIXBIT
C     FORMAT(I2,2X,10(I1,1X))                                            SIXBIT
C     SAMPLE                                                             SIXBIT
C     /07  2 1 0 0 1 0 4                                                 SIXBIT
C 9.  FLAGS AND STARTING VALUES, READ IN START. INPUTS ARE               SIXBIT
C     T - THE STARTING RUNNING TIME IN SECONDS (NORMALLY ZERO EXCEPT     SIXBIT
C       WHEN A RUN IS BEING RESTARTED)                                   SIXBIT
C     TPRINT - THE DESIRED INTERVAL, IN SECONDS, BETWEEN PRINT-OUTS.     SIXBIT
C       THIS WILL AUTOMATICALLY INCREASE LATER IF THE STEPS GET BIG      SIXBIT
C       ENOUGH.  1 SECOND IS A REASONABLE CHOICE HERE.                   SIXBIT
C     TMAX1 - THE MAXIMUM VALUE WHICH TPRINT IS ALLOWED TO HAVE WHEN     SIXBIT
C             THE ATMOSPHERE IS IN FULL DAYLIGHT OR FULL DARKNESS.       SIXBIT
C             RECOMMENDED VALUE IS 3600 SECONDS.                         SIXBIT
C     TMAX2 - THE MAXIMUM VALUE WHICH TPRINT CAN HAVE DURING TWILIGHT.   SIXBIT
C             RECOMMENDED VALUE IS 60 SECONDS.                           SIXBIT
C     TQUIT - IF T REACHES THIS VALUE, THE PROGRAM STOPS (IN SECONDS).   SIXBIT
C     IFLAG8 - =0 TO USE A RUNGE-KUTTA SOLUTION (RUNKUT)                 SIXBIT
C              =1 TO USE A PARTIAL INTEGRATION SOLUTION (WEIRD)          SIXBIT
C     IFLAG1 - =1 LETS SPECIES GO INTO EQUILIBRIUM.                      SIXBIT
C              =0 DOESNT LET THEM.                                       SIXBIT
C     EQUIL - THE VALUE OF THE FRACTIONAL CHANGE PER SECOND WHICH        SIXBIT
C       DEFINES EQUILIBRIUM.  1.0E-8 IS REASONABLE.                      SIXBIT
C     JOSHUA =0 GIVES NORMAL OPERATION.                                  SIXBIT
C            =1 HOLDS THE ZENITH ANGLE AT ITS STARTING VALUE, USEFUL     SIXBIT
C              FOR DETERMINING STEADY-STATE VALUES.                      SIXBIT
C     FORMAT (5(E8.2,2X),2(I1,2X),E8.2,2X,I1)                            SIXBIT
C     SAMPLE                                                             SIXBIT
C/0.00E+00  1.00E+00  3.60E+03  6.00E+01  8.64E+05  0  1  1.00E-08  0    SIXBIT
C                                                                        SIXBIT
C 10. LOCATION INFORMATION, READ IN ZENANG.                              SIXBIT
C     LAT = LATITUDE IN DEGREES                                          SIXBIT
```

```
C     LONG = LONGITUDE IN DEGREES                                     SIXBIT
C     TMONTH = MONTH NUMBER (INTEGER)                                 SIXBIT
C     TDAY = DAY OF THE MONTH (INTEGER)                               SIXBIT
C     THOUR = HOUR OF THE DAY (INTEGER)                               SIXBIT
C     TMIN = MINUTE OF THE HOUR (INTEGER)                             SIXBIT
C     TSEC = SECOND OF THE MINUTE (INTEGER)                           SIXBIT
C     FORMAT (2(E9.2,2X),5(I2,2X))                                    SIXBIT
C     SAMPLE                                                          SIXBIT
C     /+3.72E+01  +9.25E+01  03  05  23  00  00                       SIXBIT
C                                                                     SIXBIT
C 11. SPECIES NUMBER AND BEGINNING ALTITUDE PROFILE, READ IN START.   SIXBIT
C     PROFILE IS TAKEN TO BE ALL ZEROS IF NO PROFILE IS GIVEN.        SIXBIT
C     FORMAT (I2) - FIRST CARD - SPECIES NUMBER                       SIXBIT
C     FORMAT (7(E8.2,2X)) - FOLLOWING CARDS - ALTITUDE PROFILE        SIXBIT
C     SAMPLE                                                          SIXBIT
C     /04                                                             SIXBIT
C     /9.37E+08  4.25E+09  8.34E+09                                   SIXBIT
C                                                                     SIXBIT
C FLAG USAGE * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  SIXBIT
C     IFLAG1 =1 LETS SPECIES GO INTO EQUILIBRIUM                      SIXBIT
C            =0 DOESNT LET THEM. INPUT, SEE NO. 7.                    SIXBIT
C     IFLAG2 - USED BY RUNKUT AND WIERD TO KNOW WHEN ITS TIME TO RETURN.SIXBIT
C     IFLAG3 - USED BY ZENANG TO KNOW IF ITS THE FIRST TIME THROUGH.  SIXBIT
C     IFLAG4 AND IFLAG5 - USED BY SIXBIT TO INCREASE TPRINT. SEE NO. 7. SIXBIT
C     IFLAG6 - USED BY START TO TELL IF AN O2 PROFILE HAS BEEN INPUT. SIXBIT
C     IFLAG7 - USED TO TELL IF AN O3 PROFILE HAS BEEN INPUT.          SIXBIT
C     IFLAG8 - =0 TO USE RUNKUT                                       SIXBIT
C              =1 TO USE WEIRD. INPUT, SEE NO. 7.                     SIXBIT
C     IFLAG9 - USED BY MOST SUBROUTINES TO FIND INPUT ERRORS AND      SIXBIT
C                TERMINATE THE RUN.                                   SIXBIT
C     JOSHUA = 0 LETS THE SUN MOVE NATURALLY,                         SIXBIT
C            = 1 HOLDS THE SUN FIXED IN THE SKY. (INPUT)              SIXBIT
C     JFLAG2 = 0 FOR FRESH START,                                     SIXBIT
C            = 1 FOR RESTART (INPUT).                                 SIXBIT
C     JFLAG6 = 0 IF ALTITUDES ARE GIVEN AS TOP, BOTTOM, AND STEP SIZE, SIXBIT
C            = THE NUMBER OF ALTITUDES IF GIVEN IN TABULAR FORM. (INPUT)SIXBIT
C     JJFLAG = 0 FOR NO MICROFILM,                                    SIXBIT
C            = 1 TO GET MICROFILM PLOTS OF THE PROFILES. (INPUT)      SIXBIT
      READ (13,997) I UD                                              SIXBIT
      REWIND 13                                                       SIXBIT
      READ (60,997) LAG2                                              SIXBIT
      READ (60,997) JJFLAG
      IUNIT = 60                                                      SIXBIT
      IF (JFLAG2 .EQ. 1) IUNIT = 13                                   SIXBIT
      READ (60   ,996  (INFORM(INFO),INFO=1,30)                       SIXBIT
      PRINT 995, (INF RM(INFO),INFO=1,30)                             SIXBIT
      IFLAG9=0                                                        SIXBIT
      CALL REACTS                                                     SIXBIT
      IF (IFLAG9 .EQ. 1) GO TO 9000                                   SIXBIT
      CALL HEIGHTS                                                    SIXBIT
      IF (IFLAG9 .EQ  .) GO TO 9000                                   SIXBIT
      CALL ALGEBRA                                                    SIXBIT
      IF (IFLAG9 .EQ. .) GO TO 9000                                   SIXBIT
      CALL START                                                      SIXBIT
      IF (IFLAG9 .EQ.  ) GO TO 9000                                   SIXBIT
   10 CALL ZENANG                                                     SIXBIT
      IF (IFLAG9 .EQ.  ) GO TO 9000                                   SIXBIT
      DO 100 JZMAIN =  ,JZEND                                         SIXBIT
      IJZX = JZEND-JZ  IN                                             SIXBIT
      DO 30 M1=1,NSPE                                                 SIXBIT
```

```
      Y(M1) = YHOLD(M1,JZMAIN)                                          SIXBIT
   30 CONTINUE                                                          SIXBIT
      CALL PRODUC                                                       SIXBIT
      HCOUNT = 0                                                        SIXBIT
      IF (IFLAG8 .EQ. 1) GO TO 40                                       SIXBIT
      CALL RUNKUT                                                       SIXBIT
      GO TO 50                                                          SIXBIT
   40 CALL WEIRD                                                        SIXBIT
   50 HBAR(JZMAIN) = TPRINT/HCOUNT                                      SIXBIT
      IFLAG2 = 0                                                        SIXBIT
      DO 65 ITCH=1,ITCHXX                                               SIXBIT
      SUM = 0.0                                                         SIXBIT
      DO 55 M2=3,NSPEC                                                  SIXBIT
      SUM = SUM + Y(M2)*IATOM(ITCH,M2)                                  SIXBIT
   55 CONTINUE                                                          SIXBIT
      IF (SUM .LT. 1.0) GO TO 65                                        SIXBIT
      DO 60 M22 = 3,NSPEC                                               SIXBIT
      IF (IATOM(ITCH,M22) .EQ. 0) GO TO 60                              SIXBIT
      Y(M22) = Y(M22)*SUMIN(ITCH,JZMAIN)/SUM                            SIXBIT
   60 CONTINUE                                                          SIXBIT
      Y(2) = Y(2)*SUMIN(ITCH,JZMAIN)/SUM                                SIXBIT
   65 CONTINUE                                                          SIXBIT
      DO 70 M222=2,NSPEC                                                SIXBIT
      YHOLD(M222,JZMAIN) = Y(M222)                                      SIXBIT
   70 CONTINUE                                                          SIXBIT
      DO 73 I=1,NREAC                                                   SIXBIT
      PROHLD(I,JZMAIN) = TERM(I)                                        SIXBIT
   73 CONTINUE                                                          SIXBIT
      DO 76 II=1,NFREAC                                                 SIXBIT
      PROHLD(II+NREAC,JZMAIN) = FTERM(II)                               SIXBIT
   76 CONTINUE                                                          SIXBIT
      T = TOLD                                                          SIXBIT
  100 CONTINUE                                                          SIXBIT
      T = T + TPRINT                                                    SIXBIT
      IF (IFLAG1 .EQ. 0) GO TO 150                                      SIXBIT
C FIND OUT WHAT HAS GONE INTO EQUILIBRIUM                               SIXBIT
      DO 140 JZ1=1,JZEND                                                SIXBIT
      IF (JPULL(JZ1) .EQ. 1) GO TO 140                                  SIXBIT
      PCHECK = 0                                                        SIXBIT
      DO 130 M4=3,NSPEC                                                 SIXBIT
      PHOLD(M4,JZ1) = 0                                                 SIXBIT
      IF (YHOLD(M4,JZ1) .LT. 1.0E-50 .AND. YLAST (M4,JZ1) .LT. 1.0E-50) SIXBIT
     1     GO TO 110                                                    SIXBIT
      IF (YLAST(M4,JZ1) .LT. 1.0E-50) GO TO 130                         SIXBIT
      IF (ABS(1.0-YHOLD(M4,JZ1)/YLAST(M4,JZ1)) .LT. EQUIL*TPRINT)       SIXBIT
     1    GO TO 110                                                     SIXBIT
      GO TO 130                                                         SIXBIT
  110 PHOLD(M4,JZ1) = 1                                                 SIXBIT
  120 PCHECK = PCHECK+1                                                 SIXBIT
  130 CONTINUE                                                          SIXBIT
      IF (PCHECK+2 .LT. NSPEC) GO TO 140                                SIXBIT
      JPULL(JZ1) = 1                                                    SIXBIT
      HBAR(JZ1) = TQUIT                                                 SIXBIT
  140 CONTINUE                                                          SIXBIT
C SET UP AND PRINT OUT ALL THE SPECIES                                  SIXBIT
  150 DO 170 JZ2=1,JZEND                                                SIXBIT
      ZPRINT(JZ2) = Z(JZ2)*(1.0-2.0*JPULL(JZ2))                         SIXBIT
      YPRINT(2,JZ2) = YHOLD(2,JZ2)                                      SIXBIT
      DO 160 M5=3,NSPEC                                                 SIXBIT
      YPRINT(M5,JZ2) = YHOLD(M5,JZ2)*(1.0-2.0*PHOLD(M5,JZ2))            SIXBIT
```

```
  160 CONTINUE                                                          SIXBIT
  170 CONTINUE                                                          SIXBIT
      CALL PHYSICS                                                      SIXBIT
      CALL HANDLE                                                       SIXBIT
      IF (JJFLAG .NE. 1) GO TO 175                                      SIXBIT
      CALL MICROPLT                                                     SIXBIT
  175 TOLD = T                                                          SIXBIT
      IF (TOLD .GE. TQUIT) GO TO 800                                    SIXBIT
C ADJUST TPRINT IF NECCESSARY                                           SIXBIT
      IFLAG4 = IFLAG4+1                                                 SIXBIT
      IF (IFLAG4 .LT. 10) GO TO 188                                     SIXBIT
      IFLAG4 = 0                                                        SIXBIT
      IFLAG5 = IFLAG5+1                                                 SIXBIT
      IF (IFLAG5 .GT. 9) IFLAG5 = 0                                     SIXBIT
      HMIN = TPRINT                                                     SIXBIT
      DO 180 JZ3=1,JZEND                                                SIXBIT
      IF (HBAR(JZ3) .LT. HMIN) HMIN = HBAR(JZ3)                         SIXBIT
  180 CONTINUE                                                          SIXBIT
      IF (HMIN/TPRINT .LT. 0.001) GO TO 188                             SIXBIT
      IF (ALPHA .LT. 90.0 .OR. ALPHA .GT. 90.0+SQRTF(Z(1))) GO TO 182   SIXBIT
      IF (TPRINT*10.0 .LT. TMAX2) GO TO 184                             SIXBIT
      TPRINT = TMAX2                                                    SIXBIT
      GO TO 190                                                         SIXBIT
  182 IF (TPRINT*10.0 .LT. TMAX1) GO TO 184                             SIXBIT
      TPRINT = TMAX1                                                    SIXBIT
      GO TO 190                                                         SIXBIT
  184 TPRINT = TPRINT*10.0                                              SIXBIT
      IFLAG4 = IFLAG5                                                   SIXBIT
      IFLAG5 = 0                                                        SIXBIT
      GO TO 190                                                         SIXBIT
  188 IF (ALPHA .GT. 90.0 .AND. ALPHA .LT. 90.0+SQRTF(Z(1))             SIXBIT
     1    .AND. TPRINT .GT. TMAX2) TPRINT = TMAX2                       SIXBIT
  190 DO 210 JZ4=1,JZEND                                                SIXBIT
      DO 200 M6=3,NSPEC                                                 SIXBIT
      YLAST(M6,JZ4) = YHOLD(M6,JZ4)                                     SIXBIT
  200 CONTINUE                                                          SIXBIT
  210 CONTINUE                                                          SIXBIT
      DO 220 JZ5=1,JZEND                                                SIXBIT
      IF (JPULL(JZ5) .EQ. 0) GO TO 10                                   SIXBIT
  220 CONTINUE                                                          SIXBIT
  700 PRINT 999                                                         SIXBIT
      GO TO 9000                                                        SIXBIT
  800 PRINT 998                                                         SIXBIT
  995 FORMAT(1H0,10A8)                                                  SIXBIT
  996 FORMAT(10A8)                                                      SIXBIT
  997 FORMAT(I1)                                                        SIXBIT
  998 FORMAT (1H0,*THE RUNNING TIME HAS REACHED TQUIT, THE QUITTING TIMESIXBIT
     1 THAT WAS INPUT IN SUBROUTINE START.*)                            SIXBIT
  999 FORMAT(1H0,*ALL SPECIES AT ALL ALTITUDES ARE NOW IN EQUILIBRIUM.*)SIXBIT
 9000 END                                                               SIXBIT-
```

```
      SUBROUTINE REACTS                                                 REACTS
      COMMON /A/      A(51)            ,ARK(4)          ,B(51)        , REACTS
     1 BRK(4)        ,CRK(4)           ,F(20)           ,FF(20)       , REACTS
     2 FLUX(120)     ,FLXINF(120)      ,FRATE(20)       ,FTERM(40)    , REACTS
     3 FTERMW(40)    ,G(20)            ,GG(20)          ,HBAR(20)     , REACTS
     4 H2HOLD(20)    ,H3HOLD(20)       ,IATOM(10,20)    ,IGAIN(4)     , REACTS
     5 IGAINA(20,20) ,IGAINB(20,20)    ,IGAINC(20,20)   ,IGAIND(20,20), REACTS
     6 ILOSS(4)      ,ILOSSA(20,20)    ,ILOSSB(20,20)   ,ILOSSC(20,20), REACTS
     7 ILOSSD(20,20) ,INFORM(30)       ,JPULL(20)       ,K(1000)      , REACTS
     8 K0(51)        ,KABS(20,120)     ,KABSN2(120)     ,KABSO2(120)  , REACTS
     9 KABSO3(120)   ,KRK(80)          ,L1(51)          ,L2(51)       , REACTS
     X L3(51)        ,LL1(21)          ,LL2(21)         ,LL3(21)      , REACTS
     1 N(51)         ,NASRAS(1000)     ,NFTERM(20)      ,NTERM(20)    , REACTS
     2 NUSRUS(400)   ,PHOLD(20,20)     ,PROHLD(71,20)   ,QOLD(20)     , REACTS
     3 QRK(20)       ,R1(51)           ,R2(51)          ,R3(51)       , REACTS
     4 RR1(21)       ,RR2(21)          ,RR3(21)         ,SUMIN(10,20) , REACTS
     5 TABO2(87)     ,TABO3(87)        ,TEMP(20)        ,TERM(100)    , REACTS
     6 TERMW(100)    ,Y(20)            ,YHOLD(20,20)    ,YLAST(20,20) , REACTS
     7 YNEW(20)      ,YOLD(20)         ,YPRINT(20,20)   ,YSTART(20,20), REACTS
     8 Y3SUN(20)     ,Y4SUN(20)        ,Z(20)           ,ZPRINT(20)     REACTS
      COMMON /B/  ALPHA , EQUIL , H     , HCOUNT, IFLAG1, IFLAG2,       REACTS
     2            IFLAG3, IFLAG4, IFLAG5, IFLAG6, IFLAG7, IFLAG8,       REACTS
     3            IFLAG9, IJZX  , ITCHXX, IUNIT , JCNTRL, JFLAG2,       REACTS
     4            JFLAG6, JOSHUA, JZEND , JZMAIN, LAT   , LONG  ,       REACTS
     5            NFREAC, NREAC , NSPEC , PI    , T     , TDAY  ,       REACTS
     6            THOUR , TMAX1 , TMAX2 , TMIN  , TMONTH, TOLD  ,       REACTS
     7            TPRINT, TQUIT , TSEC  , ZBOTUM, ZSTEP , ZTOP          REACTS
      INTEGER HCOUNT,PHOLD,R1,R2,R3,RR1,RR2,RR3,RX1,RX2,RX3,            REACTS
     1 TMONTH,TDAY,THOUR,TMIN,TSEC,TX                                   REACTS
      REAL K0,N,KABS,K,KRK,LAT,LONG,K5,K5MAX,KABSO2,KABSO3,KABSN2       REACTS
      NSPEC = 0                                                         REACTS
      DO 10 I1=1,51                                                     REACTS
      READ (60    ,999) L1(I1),L2(I1),L3(I1),R1(I1),R2(I1),R3(I1),      REACTS
     1           K0(I1),A(I1),N(I1),B(I1)                               REACTS
      IF (L1(I1) .EQ. 99) GO TO 20                                      REACTS
      IXX =XMAXOF(L1(I1),L2(I1),L3(I1),R1(I1),R2(I1),R3(I1))            REACTS
      IF (IXX .GT. NSPEC) NSPEC = IXX                                   REACTS
   10 CONTINUE                                                          REACTS
      PRINT 998                                                         REACTS
      IFLAG9 = 1                                                        REACTS
      GO TO 900                                                         REACTS
   20 NREAC = I1-1                                                      REACTS
      NSPEC = NSPEC-2                                                   REACTS
      PRINT 997,NREAC,NSPEC                                             REACTS
      PRINT 991, (I3,L1(I3),L2(I3),L3(I3),R1(I3),R2(I3),R3(I3),         REACTS
     1           K0(I3),A(I3),N(I3),B(I3),I3=1,NREAC)                   REACTS
      NSPEC = NSPEC+2                                                   REACTS
      DO 30 I2=1,21                                                     REACTS
      READ (60    ,996) LL1(I2),LL2(I2),LL3(I2),RR1(I2),RR2(I2),RR3(I2) REACTS
      IF (LL1(I2) .EQ. 99) GO TO 40                                     REACTS
      READ (60,992) LAMDA1,LAMDA2                                       REACTS
      READ (60    ,995) (KABS(I2,LAMDA),LAMDA=LAMDA1,LAMDA2)            REACTS
   30 CONTINUE                                                          REACTS
      PRINT 994                                                         REACTS
```

```
      IFLAG9 = 1                                                        REACTS
      GO TO 900                                                         REACTS
   40 NFREAC = I2-1                                                     REACTS
      PRINT 993,NFREAC                                                  REACTS
      DO 50 I5 = 1,NFREAC                                               REACTS
      I4 = I5+NREAC                                                     REACTS
      PRINT 990, I4,LL1(I5),RR1(I5),RR2(I5),RR3(I5)                     REACTS
   50 CONTINUE                                                          REACTS
  999 FORMAT(3(I2,1X),1X,3(I2,1X),6X,2(E8.2,2X),2(E9.2,2X))             REACTS
  998 FORMAT(1H ,*NO END CARD FOUND WHEN READING THE REACTION SET IN SUBREACTS
     1ROUTINE REACTS.*,/,*THIS WILL TERMINATE THE RUN. SORRY.*)         REACTS
  997 FORMAT(1H ,*THERE HAVE BEEN *,I2,* REACTIONS INPUT INVOLVING *,I2,REACTS
     1* SPECIES.*,/1X*THE REACTIONS AND THEIR RATE COEFFICIENTS ARE *,/)REACTS
  996 FORMAT (3(I2,1X),1X,3(I2,1X))                                     REACTS
  995 FORMAT(7(E8.2,2X))                                                REACTS
  994 FORMAT(1H ,*NO END CARD FOUND WHEN READING THE PHOTOREACTIONS IN SREACTS
     1UBROUTINE REACTS.*,/,*THIS WILL TERMINATE THE RUN. TOUGH LUCK.*)  REACTS
  993 FORMAT(1H0,*THERE HAVE BEEN *,I2,* PHOTOREACTIONS INPUT.*)        REACTS
  992 FORMAT(I3,2X,I3)                                                  REACTS
  991 FORMAT(1H ,I2,3X,2(I2,*+*),I2,*---*,2(I2,*+*),I2,*    K = *,E8.2, REACTS
     17H*(TEMP/,E8.2,3H)**,F5.2,6H*EXPF(,E9.2,*/TEMP)*,///)             REACTS
  990 FORMAT (1H ,I2,4X,I2,*+ PHOTON ---*,2(I2,*+*),I2,///)             REACTS
  900 RETURN                                                            REACTS
      END                                                               REACTS-
```

```
      SUBROUTINE HEIGHTS                                                HEIGHTS
      COMMON /A/       A(51)            ,ARK(4)          ,B(51)         , HEIGHTS
     1 BRK(4)          ,CRK(4)          ,F(20)           ,FF(20)        , HEIGHTS
     2 FLUX(120)       ,FLXINF(120)     ,FRATE(20)       ,FTERM(40)     , HEIGHTS
     3 FTERMW(40)      ,G(20)           ,GG(20)          ,HBAR(20)      , HEIGHTS
     4 H2HOLD(20)      ,H3HOLD(20)      ,IATOM(10,20)    ,IGAIN(4)      , HEIGHTS
     5 IGAINA(20,20)   ,IGAINB(20,20)   ,IGAINC(20,20)   ,IGAIND(20,20) , HEIGHTS
     6 ILOSS(4)        ,ILOSSA(20,20)   ,ILOSSB(20,20)   ,ILOSSC(20,20) , HEIGHTS
     7 ILOSSD(20,20)   ,INFORM(30)      ,JPULL(20)       ,K(1000)       , HEIGHTS
     8 K0(51)          ,KABS(20,120)    ,KABSN2(120)     ,KABSO2(120)   , HEIGHTS
     9 KABSO3(120)     ,KRK(80)         ,L1(51)          ,L2(51)        , HEIGHTS
     X L3(51)          ,LL1(21)         ,LL2(21)         ,LL3(21)       , HEIGHTS
     1 N(51)           ,NASRAS(1000)    ,NFTERM(20)      ,NTERM(20)     , HEIGHTS
     2 NUSRUS(400)     ,PHOLD(20,20)    ,PROHLD(71,20)   ,QOLD(20)      , HEIGHTS
     3 QRK(20)         ,R1(51)          ,R2(51)          ,R3(51)        , HEIGHTS
     4 RR1(21)         ,RR2(21)         ,RR3(21)         ,SUMIN(10,20)  , HEIGHTS
     5 TABO2(87)       ,TABO3(87)       ,TEMP(20)        ,TERM(100)     , HEIGHTS
     6 TERMW(100)      ,Y(20)           ,YHOLD(20,20)    ,YLAST(20,20)  , HEIGHTS
     7 YNEW(20)        ,YOLD(20)        ,YPRINT(20,20)   ,YSTART(20,20) , HEIGHTS
     8 Y3SUN(20)       ,Y4SUN(20)       ,Z(20)           ,ZPRINT(20)      HEIGHTS
      COMMON /B/   ALPHA , EQUIL , H     , HCOUNT, IFLAG1, IFLAG2,          HEIGHTS
     2             IFLAG3, IFLAG4, IFLAG5, IFLAG6, IFLAG7, IFLAG8,          HEIGHTS
     3             IFLAG9, IJZX  , ITCHXX, IUNIT , JCNTRL, JFLAG2,          HEIGHTS
     4             JFLAG6, JOSHUA, JZEND , JZMAIN, LAT   , LONG  ,          HEIGHTS
     5             NFREAC, NREAC , NSPEC , PI    , T     , TDAY  ,          HEIGHTS
     6             THOUR , TMAX1 , TMAX2 , TMIN  , TMONTH, TOLD  ,          HEIGHTS
     7             TPRINT, TQUIT , TSEC  , ZBOTUM, ZSTEP , ZTOP             HEIGHTS
      INTEGER HCOUNT,PHOLD,R1,R2,R3,RR1,RR2,RR3,RX1,RX2,RX3,                HEIGHTS
     1 TMONTH,TDAY,THOUR,TMIN,TSEC,TX                                       HEIGHTS
      REAL K0,N,KABS,K,KRK,LAT,LONG,K5,K5MAX,KABSO2,KABSO3,KABSN2           HEIGHTS
      READ (60    ,999) JFLAG6,(Z(JZ1),JZ1=1,6)                             HEIGHTS
      IF (JFLAG6 .EQ. 0) GO TO 100                                          HEIGHTS
      JZEND = JFLAG6                                                        HEIGHTS
      IF (JZEND .LT. 7) GO TO 200                                           HEIGHTS
      READ (60,997) (Z(JZ2),JZ2=7,JZEND)                                    HEIGHTS
      GO TO 200                                                             HEIGHTS
  100 ZTOP = Z(1)                                                           HEIGHTS
      ZBOTUM = Z(2)                                                         HEIGHTS
      ZSTEP = Z(3)                                                          HEIGHTS
      JZEND = (ZTOP-ZBOTUM)/ZSTEP + 1                                       HEIGHTS
      DO 110 JZ3=1,JZEND                                                    HEIGHTS
      Z(JZ3) = ZTOP - (JZ3-1)*ZSTEP                                         HEIGHTS
  110 CONTINUE                                                              HEIGHTS
      IF (Z(JZEND) .EQ. ZBOTUM) GO TO 200                                   HEIGHTS
      PRINT 998                                                             HEIGHTS
      IFLAG9 = 1                                                            HEIGHTS
      GO TO 9000                                                            HEIGHTS
  200 DO 210 JZ4=1,JZEND                                                    HEIGHTS
      ZPRINT(JZ4) = Z(JZ4)                                                  HEIGHTS
  210 CONTINUE                                                              HEIGHTS
  997 FORMAT(6(E9.3,2X))                                                    HEIGHTS
  998 FORMAT(1HO,*ALTITUDES DONT CHECK OUT IN SUBROUTINE HEIGHTS. THIS WHEIGHTS
     1ILL TERMINATE THE RUN.*)                                              HEIGHTS
  999 FORMAT(I2,6(2X,E9.3))                                                 HEIGHTS
 9000 RETURN                                                                HEIGHTS
      END                                                                   HEIGHT-
```

```
      SUBROUTINE ALGEBRA                                                   ALGEBRA
      COMMON /A/      A(51)            ,ARK(4)           ,B(51)          , ALGEBRA
     1 BRK(4)        ,CRK(4)           ,F(20)            ,FF(20)         , ALGEBRA
     2 FLUX(120)     ,FLXINF(120)      ,FRATE(20)        ,FTERM(40)      , ALGEBRA
     3 FTERMW(40)    ,G(20)            ,GG(20)           ,HBAR(20)       , ALGEBRA
     4 H2HOLD(20)    ,H3HOLD(20)       ,IATOM(10,20)     ,IGAIN(4)       , ALGEBRA
     5 IGAINA(20,20) ,IGAINB(20,20)    ,IGAINC(20,20)    ,IGAIND(20,20)  , ALGEBRA
     6 ILOSS(4)      ,ILOSSA(20,20)    ,ILOSSB(20,20)    ,ILOSSC(20,20)  , ALGEBRA
     7 ILOSSD(20,20) ,INFORM(30)       ,JPULL(20)        ,K(1000)        , ALGEBRA
     8 K0(51)        ,KABS(20,120)     ,KABSN2(120)      ,KABSO2(120)    , ALGEBRA
     9 KABSO3(120)   ,KRK(80)          ,L1(51)           ,L2(51)         , ALGEBRA
     X L3(51)        ,LL1(21)          ,LL2(21)          ,LL3(21)        , ALGEBRA
     1 N(51)         ,NASRAS(1000)     ,NFTERM(20)       ,NTERM(20)      , ALGEBRA
     2 NUSRUS(400)   ,PHOLD(20,20)     ,PROHLD(71,20)    ,QOLD(20)       , ALGEBRA
     3 QRK(20)       ,R1(51)           ,R2(51)           ,R3(51)         , ALGEBRA
     4 RR1(21)       ,RR2(21)          ,RR3(21)          ,SUMIN(10,20)   , ALGEBRA
     5 TABO2(87)     ,TABO3(87)        ,TEMP(20)         ,TERM(100)      , ALGEBRA
     6 TERMW(100)    ,Y(20)            ,YHOLD(20,20)     ,YLAST(20,20)   , ALGEBRA
     7 YNEW(20)      ,YOLD(20)         ,YPRINT(20,20)    ,YSTART(20,20)  , ALGEBRA
     8 Y3SUN(20)     ,Y4SUN(20)        ,Z(20)            ,ZPRINT(20)       ALGEBRA
      COMMON /B/  ALPHA , EQUIL , H     , HCOUNT, IFLAG1, IFLAG2,          ALGEBRA
     2            IFLAG3, IFLAG4, IFLAG5, IFLAG6, IFLAG7, IFLAG8,          ALGEBRA
     3            IFLAG9, IJZX  , ITCHXX, IUNIT , JCNTRL, JFLAG2,          ALGEBRA
     4            JFLAG6, JOSHUA, JZEND , JZMAIN, LAT   , LONG  ,          ALGEBRA
     5            NFREAC, NREAC , NSPEC , PI    , T     , TDAY  ,          ALGEBRA
     6            THOUR , TMAX1 , TMAX2 , TMIN  , TMONTH, TOLD  ,          ALGEBRA
     7            TPRINT, TQUIT , TSEC  , ZBOTUM, ZSTEP , ZTOP             ALGEBRA
      INTEGER HCOUNT,PHOLD,R1,R2,R3,RR1,RR2,RR3,RX1,RX2,RX3,               ALGEBRA
     1 TMONTH,TDAY,THOUR,TMIN,TSEC,TX                                      ALGEBRA
      REAL K0,N,KABS,K,KRK,LAT,LONG,K5,K5MAX,KABSO2,KABSO3,KABSN2          ALGEBRA
      READ (60    ,999) (TEMP(JZ1),JZ1=1,JZEND)                            ALGEBRA
      DO 10 I1=1,NREAC                                                     ALGEBRA
      IXX = (I1-1)*JZEND                                                   ALGEBRA
      DO 10 JZ2=1,JZEND                                                    ALGEBRA
      IJZ = IXX + JZ2                                                      ALGEBRA
      K(IJZ) = K0(I1)*(TEMP(JZ2)/A(I1))**N(I1)*EXPF(B(I1)/TEMP(JZ2))       ALGEBRA
      PROHLD(I1,JZ2) = K(IJZ)                                              ALGEBRA
   10 CONTINUE                                                             ALGEBRA
      DO 20 M1=1,NSPEC                                                     ALGEBRA
      NTERM(M1) = 0                                                        ALGEBRA
   20 CONTINUE                                                             ALGEBRA
      DO 30 IZERO = 1,1000                                                 ALGEBRA
      NASRAS(IZERO) = 0                                                    ALGEBRA
   30 CONTINUE                                                             ALGEBRA
      DO 260 I2=1,NREAC                                                    ALGEBRA
      LX1 = L1(I2)                                                         ALGEBRA
      LX2 = L2(I2)                                                         ALGEBRA
      LX3 = L3(I2)                                                         ALGEBRA
      RX1 = R1(I2)                                                         ALGEBRA
      RX2 = R2(I2)                                                         ALGEBRA
      RX3 = R3(I2)                                                         ALGEBRA
      IF (LX1 .LT. 3) GO TO 130                                            ALGEBRA
      IF (RX1 .NE. LX1) GO TO 110                                          ALGEBRA
      LX1 = 1                                                              ALGEBRA
```

```
      RX1 = 1                                                         ALGEBRA
      GO TO 130                                                       ALGEBRA
  110 IF (RX2 .NE. LX1) GO TO 120                                     ALGEBRA
      LX1 = 1                                                         ALGEBRA
      RX2 = 1                                                         ALGEBRA
      GO TO 130                                                       ALGEBRA
  120 IF (RX3 .NE. LX1) GO TO 130                                     ALGEBRA
      LX1 = 1                                                         ALGEBRA
      RX3 = 1                                                         ALGEBRA
  130 IF (LX2 .LT. 3) GO TO 160                                       ALGEBRA
      IF (RX1 .NE. LX2) GO TO 140                                     ALGEBRA
      LX2 = 1                                                         ALGEBRA
      RX1 = 1                                                         ALGEBRA
      GO TO 160                                                       ALGEBRA
  140 IF (RX2 .NE. LX2) GO TO 150                                     ALGEBRA
      LX2 = 1                                                         ALGEBRA
      RX2 = 1                                                         ALGEBRA
      GO TO 160                                                       ALGEBRA
  150 IF (RX3 .NE. LX2) GO TO 160                                     ALGEBRA
      LX2 = 1                                                         ALGEBRA
      RX3 = 1                                                         ALGEBRA
  160 IF (LX3 .LT. 3) GO TO 200                                       ALGEBRA
      IF (RX1 .NE. LX3) GO TO 170                                     ALGEBRA
      LX3 = 1                                                         ALGEBRA
      RX1 = 1                                                         ALGEBRA
      GO TO 200                                                       ALGEBRA
  170 IF (RX2 .NE. LX3) GO TO 180                                     ALGEBRA
      LX3 = 1                                                         ALGEBRA
      RX2 = 1                                                         ALGEBRA
      GO TO 200                                                       ALGEBRA
  180 IF (RX3 .NE. LX3) GO TO 200                                     ALGEBRA
      LX3 = 1                                                         ALGEBRA
      RX3 = 1                                                         ALGEBRA
  200 IF (LX1 .LT. 3) GO TO 210                                       ALGEBRA
      NTERM(LX1) = NTERM(LX1)+1                                       ALGEBRA
      NASRAS((LX1-3)*NREAC+NTERM(LX1)) = I2                           ALGEBRA
  210 IF (LX2 .LT. 3) GO TO 220                                       ALGEBRA
      NTERM(LX2) = NTERM(LX2)+1                                       ALGEBRA
      NASRAS((LX2-3)*NREAC+NTERM(LX2)) = I2                           ALGEBRA
  220 IF (LX3 .LT. 3) GO TO 230                                       ALGEBRA
      NTERM(LX3) = NTERM(LX3)+1                                       ALGEBRA
      NASRAS((LX3-3)*NREAC+NTERM(LX3)) = I2                           ALGEBRA
  230 IF (RX1 .LT. 3) GO TO 240                                       ALGEBRA
      NTERM(RX1) = NTERM(RX1)+1                                       ALGEBRA
      NASRAS((RX1-3)*NREAC+NTERM(RX1)) = I2+NREAC                     ALGEBRA
  240 IF (RX2 .LT. 3) GO TO 250                                       ALGEBRA
      NTERM(RX2) = NTERM(RX2)+1                                       ALGEBRA
      NASRAS((RX2-3)*NREAC+NTERM(RX2)) = I2+NREAC                     ALGEBRA
  250 IF (RX3 .LT. 3) GO TO 260                                       ALGEBRA
      NTERM(RX3) = NTERM(RX3)+1                                       ALGEBRA
      NASRAS((RX3-3)*NREAC+NTERM(RX3)) = I2+NREAC                     ALGEBRA
  260 CONTINUE                                                        ALGEBRA
      DO 310 IIZERO=1,400                                             ALGEBRA
      NUSRUS(IIZERO) = 0                                              ALGEBRA
```

```
310 CONTINUE                                                              ALGEBRA
    DO 320 M2=1,NSPEC                                                     ALGEBRA
    NFTERM(M2)=0                                                          ALGEBRA
320 CONTINUE                                                              ALGEBRA
    DO 350 II1=1,NFREAC                                                   ALGEBRA
    NFTERM(LL1(II1)) = NFTERM(LL1(II1)) + 1                               ALGEBRA
    NUSRUS((LL1(II1)-3)*NFREAC+NFTERM(LL1(II1))) = II1                    ALGEBRA
    IF (RR1(II1) .LT. 3) GO TO 330                                        ALGEBRA
    NFTERM(RR1(II1)) = NFTERM(RR1(II1)) + 1                               ALGEBRA
    NUSRUS((RR1(II1)-3)*NFREAC+NFTERM(RR1(II1))) = II1+NFREAC             ALGEBRA
330 IF (RR2(II1) .LT. 3) GO TO 340                                        ALGEBRA
    NFTERM(RR2(II1)) = NFTERM(RR2(II1)) + 1                               ALGEBRA
    NUSRUS((RR2(II1)-3)*NFREAC+NFTERM(RR2(II1))) = II1+NFREAC             ALGEBRA
340 IF (RR3(II1) .LT. 3) GO TO 350                                        ALGEBRA
    NFTERM(RR3(II1)) = NFTERM(RR3(II1)) + 1                               ALGEBRA
    NUSRUS((RR3(II1)-3)*NFREAC+NFTERM(RR3(II1))) = II1+NFREAC             ALGEBRA
350 CONTINUE                                                              ALGEBRA
999 FORMAT (7(E8.2,2X))                                                   ALGEBRA
    RETURN                                                                ALGEBRA
    END                                                                   ALGEBR-
```

```
      SUBROUTINE START                                                  START
      COMMON /A/       A(51)            ,ARK(4)           ,B(51)           , START
     1 BRK(4)         ,CRK(4)           ,F(20)            ,FF(20)          , START
     2 FLUX(120)      ,FLXINF(120)      ,FRATE(20)        ,FTERM(40)       , START
     3 FTERMW(40)     ,G(20)            ,GG(20)           ,HBAR(20)        , START
     4 H2HOLD(20)     ,H3HOLD(20)       ,IATOM(10,20)     ,IGAIN(4)        , START
     5 IGAINA(20,20)  ,IGAINB(20,20)    ,IGAINC(20,20)    ,IGAIND(20,20)   , START
     6 ILOSS(4)       ,ILOSSA(20,20)    ,ILOSSB(20,20)    ,ILOSSC(20,20)   , START
     7 ILOSSD(20,20)  ,INFORM(30)       ,JPULL(20)        ,K(1000)         , START
     8 KO(51)         ,KABS(20,120)     ,KABSN2(120)      ,KABSO2(120)     , START
     9 KABSO3(120)    ,KRK(80)          ,L1(51)           ,L2(51)          , START
     X L3(51)         ,LL1(21)          ,LL2(21)          ,LL3(21)         , START
     1 N(51)          ,NASRAS(1000)     ,NFTERM(20)       ,NTERM(20)       , START
     2 NUSRUS(400)    ,PHOLD(20,20)     ,PROHLD(71,20)    ,OOLD(20)        , START
     3 QRK(20)        ,R1(51)           ,R2(51)           ,R3(51)          , START
     4 RR1(21)        ,RR2(21)          ,RR3(21)          ,SUMIN(10,20)    , START
     5 TABO2(87)      ,TABO3(87)        ,TEMP(20)         ,TERM(100)       , START
     6 TERMW(100)     ,Y(20)            ,YHOLD(20,20)     ,YLAST(20,20)    , START
     7 YNEW(20)       ,YOLD(20)         ,YPRINT(20,20)    ,YSTART(20,20)   , START
     8 Y3SUN(20)      ,Y4SUN(20)        ,Z(20)            ,ZPRINT(20)        START
      COMMON /B/  ALPHA , EOUIL , H     , HCOUNT, IFLAG1, IFLAG2,          START
     2            IFLAG3, IFLAG4, IFLAG5, IFLAG6, IFLAG7, IFLAG8,          START
     3            IFLAG9, IJZX  , ITCHXX, IUNIT , JCNTRL, JFLAG2,          START
     4            JFLAG6, JOSHUA, JZEND , JZMAIN, LAT   , LONG  ,          START
     5            NFREAC, NREAC , NSPEC , PI    , T     , TDAY  ,          START
     6            THOUR , TMAX1 , TMAX2 , TMIN  , TMONTH, TOLD  ,          START
     7            TPRINT, TQUIT , TSEC  , ZBOTUM, ZSTEP , ZTOP             START
      INTEGER HCOUNT,PHOLD,R1,R2,R3,RR1,RR2,RR3,RX1,RX2,RX3,               START
     1 TMONTH,TDAY,THOUR,TMIN,TSEC,TX                                      START
      REAL KO,N,KABS,K,KRK,LAT,LONG,K5,K5MAX,KABSO2,KABSO3,KABSN2          START
      DATA (IFLAG2=0),(IFLAG3=0),(IFLAG6=0),(IFLAG7=0),(PI=3.141592654),START
     1 ((ARK(I),I=1,4)=0.5,0.292893,1.707107,0.166666667),                 START
     2 ((BRK(I),I=1,4)=2.0,1.0,1.0,2.0),                                   START
     3 ((CRK(I),I=1,4)=0.5,0.292893,1.707107,0.5),                         START
     4 ((HBAR(JZ),JZ=1,20) = 20(1.0)),                                     START
     5 ((JPULL(JZ),JZ=1,20) = 20(0)),                                      START
     6 (((YHOLD(M,JZ),M=1,2),JZ=1,20) = 40(1.0)),                          START
     7 (((YPRINT(M,JZ),M=1,2),JZ=1,20) = 40(1.0)),                         START
     8 (((PHOLD(M,JZ),M=1,2),JZ=1,20) = 40(1)),                            START
     9 (((YHOLD(M,JZ),M=3,20),JZ=1,20) = 360(0.0)),                        START
     1 (((SUMIN(ITCH,JZ),ITCH=1,10),JZ=1,20) = 200(0)),                    START
     2 (((PHOLD(M,JZ),M=3,20),JZ=1,20) = 360(0))                           START
      DATA ((FLXINF(LAMDA),LAMDA=1,60) =                                   START
     1 3.79E08, 2.65E09, 3.80E09, 8.83E09, 3.52E09, 7.71E09, 5.08E09,      START
     2 5.88E09, 1.20E10, 2.16E10, 1.75E10, 1.03E10, 3.94E11, 2.72E10,      START
     3 7.30E10, 2.54E11, 1.03E12, 1.59E12, 3.74E12, 8.88E12, 1.64E13,      START
     4 3.58E13, 5.68E13, 7.10E13, 9.90E13, 1.45E14, 2.21E14, 3.12E14,      START
     5 6.12E14, 9.38E14, 1.09E15, 1.27E15, 1.49E15, 1.83E15, 2.05E15,      START
     6 2.12E15, 2.23E15, 2.40E15, 2.36E15, 3.06E15, 3.53E15, 3.93E15,      START
     7 4.25E15, 4.45E15, 4.73E15, 4.94E15, 5.00E15, 5.20E15, 5.20E15,      START
     8 5.20E15, 5.20E15, 5.20E15, 5.30E15, 5.30E15, 5.40E15, 5.40E15,      START
     9 5.40E15, 5.40E15, 5.40E15, 5.40E15)                                 START
      DATA ((FLXINF(LAMDA),LAMDA=61,120) =                                 START
     1 5.40E15, 5.40E15, 5.40E15, 5.35E15, 5.35E15, 5.30E15, 5.25E15,      START
```

```
2 5.20E15, 5.15E15, 5.10E15, 5.05E15, 5.00E15, 4.95E15, 4.90E15,    START
3 4.85E15, 4.80E15, 4.72E15, 4.64E15, 4.56E15, 4.48E15, 4.44E15,    START
4 4.39E15, 4.34E15, 4.30E15, 4.25E15, 4.21E15, 4.17E15, 4.13E15,    START
5 4.09E15, 4.05E15, 4.00E15, 3.95E15, 3.90E15, 3.85E15, 3.80E15,    START
6 3.77E15, 3.74E15, 3.71E15, 3.68E15, 3.65E15, 3.62E15, 3.58E15,    START
7 3.54E15, 3.50E15, 3.46E15, 3.42E15, 3.39E15, 3.36E15, 3.33E15,    START
8 3.30E15, 3.26E15, 3.22E15, 3.18E15, 3.14E15, 3.10E15, 3.06E15,    START
9 3.01E15, 2.97E15, 2.93E15, 2.88E15)                               START
 DATA ((KABSN2(LAMDA),LAMDA=1,60) =                                 START
1 5.0E-18, 1.1E-17, 1.2E-17, 1.4E-17, 1.5E-17, 1.6E-17, 1.8E-17,    START
2 1.9E-17, 1.0E-18, 1.0E-18, 1.0E-18, 2.0E-21, 2.0E-21, 2.0E-21,    START
3 2.0E-21, 45(0.0E-00))                                             START
 DATA ((KABSN2(LAMDA),LAMDA=61,120) =                               START
1 60(0.0E-00))                                                      START
 DATA ((KABSO3(LAMDA),LAMDA=1,60) =                                 START
1 1.5E-20, 3.0E-20, 5.0E-20, 9.0E-20, 1.7E-19, 3.2E-19, 6.0E-19,    START
2 1.0E-18, 2.0E-18, 3.6E-18, 7.0E-18, 1.0E-17, 1.0E-17, 1.0E-17,    START
3 3.6E-18, 1.6E-18, 8.3E-19, 6.0E-19, 4.5E-19, 3.2E-19, 5.0E-19,    START
4 1.0E-18, 2.3E-18, 5.0E-18, 9.0E-18, 1.0E-17, 5.0E-18, 2.3E-18,    START
5 1.3E-18, 7.0E-19, 2.0E-19, 8.0E-20, 2.0E-20, 1.0E-21, 1.0E-21,    START
6 1.0E-21, 0.0E-00, 0.0E-00, 0.0E-00, 0.0E-00, 0.0E-00, 0.0E-00,    START
7 0.0E-00, 0.0E-00, 1.0E-24, 1.3E-22, 2.0E-22, 3.3E-22, 5.0E-22,    START
8 8.0E-22, 1.1E-21, 1.3E-21, 1.5E-21, 1.8E-21, 2.1E-21, 2.5E-21,    START
9 2.9E-21, 3.4E-21, 4.0E-21, 4.7E-21)                               START
 DATA ((KABSO3(LAMDA),LAMDA=61,120) =                               START
1 4.8E-21, 4.0E-21, 3.3E-21, 2.8E-21, 2.4E-21, 2.0E-21, 1.7E-21,    START
2 1.5E-21, 1.2E-21, 1.1E-21, 7.8E-22, 5.0E-22, 3.2E-22, 2.0E-22,    START
3 46(0.0E-00))                                                      START
 DATA ((KABSO2(LAMDA),LAMDA=1,120) =                                START
1 6.0E-19, 6.0E-18, 1.4E-17, 2.0E-17, 2.3E-17, 2.3E-17, 2.1E-17,    START
2 1.5E-17, 9.0E-18, 3.8E-18, 1.1E-18, 3.4E-19, 1.0E-20, 7.2E-18,    START
3 1.3E-17, 8.8E-18, 3.3E-18, 5.3E-19, 1.8E-20, 2.8E-23, 1.6E-23,    START
4 1.0E-23, 6.6E-24, 4.7E-24, 1.5E-24, 5.0E-25, 94(0.0))             START
 DATA ((TABO2(JZTAB),JZTAB=1,87) =                                  START
1 1.61E09, 1.70E09, 1.79E09, 1.88E09, 1.98E09, 2.08E09, 2.20E09,    START
2 2.32E09, 2.46E09, 2.60E09, 2.74E09, 2.89E09, 3.04E09, 3.22E09,    START
3 3.41E09, 3.60E09, 3.81E09, 4.02E09, 4.26E09, 4.52E09, 4.78E09,    START
4 5.15E09, 5.52E09, 6.00E09, 6.63E09, 7.26E09, 8.15E09, 9.04E09,    START
5 1.01E10, 1.16E10, 1.31E10, 1.53E10, 1.75E10, 2.04E10, 2.50E10,    START
6 2.96E10, 3.80E10, 4.64E10, 6.00E10, 8.01E10, 1.03E11, 1.38E11,    START
7 1.74E11, 2.26E11, 3.12E11, 3.98E11, 5.61E11, 7.24E11, 1.08E12,    START
8 1.54E12, 2.00E12, 3.06E12, 4.14E12, 6.04E12, 9.50E12, 1.30E13,    START
9 1.90E13, 2.76E13, 3.98E13, 5.70E13, 8.06E13, 1.02E14, 1.54E14,    START
1 2.10E14, 2.78E14, 3.68E14, 4.78E14, 6.16E14, 7.86E14, 9.98E14,    START
2 1.27E15, 1.60E15, 2.02E15, 2.58E15, 3.30E15, 4.26E15, 5.52E15,    START
3 7.22E15, 9.52E15, 1.27E16, 1.69E16, 2.28E16, 3.06E16, 3.80E16,    START
4 5.64E16, 7.66E16, 7.66E16)                                        START
 DATA ((TABO3(JZTAB),JZTAB=1,87) =                                  START
1 1.00E00, 1.30E00, 1.90E00, 2.40E00, 3.10E00, 4.00E00, 6.00E00,    START
2 8.00E00, 1.05E01, 1.40E01, 1.95E01, 2.60E01, 3.50E01, 4.50E01,    START
3 6.20E01, 9.00E01, 1.30E02, 1.80E02, 2.40E02, 3.40E02, 4.80E02,    START
4 7.00E02, 1.10E03, 1.70E03, 2.20E03, 3.20E03, 5.00E03, 7.20E03,    START
5 1.00E04, 1.50E04, 2.00E04, 3.00E04, 4.00E04, 5.80E04, 8.00E04,    START
6 1.01E05, 1.60E05, 2.10E05, 3.00E05, 3.90E05, 5.00E05, 7.00E05,    START
```

```
     7 9.20E05, 1.20E06, 1.70E06, 2.00E06, 2.60E06, 3.50E06, 4.50E06,    START
     8 5.70E06, 7.50E06, 9.00E06, 1.10E07, 1.30E07, 1.70E07, 2.10E07,    START
     9 2.60E07, 3.50E07, 4.80E07, 7.00E07, 1.00E08, 1.40E08, 1.90E08,    START
     1 2.90E08, 4.00E08, 5.60E08, 8.50E08, 1.30E09, 2.00E09, 2.90E09,    START
     2 4.10E09, 6.00E09, 1.00E10, 1.50E10, 2.20E10, 3.60E10, 5.60E10,    START
     3 8.50E10, 1.40E11, 2.50E11, 6.00E11, 1.40E12, 3.00E12, 4.00E12,    START
     4 5.00E12, 6.00E12, 6.00E12)                                        START
      IFLAG4 = 0                                                         START
      IFLAG5 = 0                                                         START
      DO 10 M1=3,NSPEC                                                   START
      READ (60,999) M2, (IATOM(ITCH,M2),ITCH=1,10)                       START
   10 CONTINUE                                                           START
      READ (60,994) T,TPRINT,TMAX1,TMAX2,TQUIT,IFLAG8,                   START
     1             IFLAG1,EQUIL,JOSHUA                                   START
      READ (60,993) LAT,LONG,TMONTH,TDAY,THOUR,TMIN,TSEC                 START
      DO 20 M3=2,NSPEC                                                   START
      READ (IUNIT,998) M4                                                START
      IF (M4 .EQ. 3) IFLAG6=1                                            START
      IF (M4 .EQ. 4) IFLAG7=1                                            START
      IF (M4 .EQ. 99) GO TO 30                                           START
      YNEW(M4) = 2.0                                                     START
      READ (IUNIT,997) (YHOLD(M4,JZ2),JZ2=1,JZEND)                       START
   20 CONTINUE                                                           START
      PRINT 996                                                          START
      IFLAG9 = 1                                                         START
      GO TO 9000                                                         START
   30 IF (IUNIT .EQ. 13) REWIND 13                                       START
      IF (IFLAG6 .EQ. 1 .AND. IFLAG7 .EQ. 1) GO TO 40                    START
      IF (Z(1) .LE. 200.0 .AND. Z(JZEND) .GE. 30.0) GO TO 40             START
      PRINT 995                                                          START
      IFLAG9 = 1                                                         START
      GO TO 9000                                                         START
   40 DO 60 JZ3=1,JZEND                                                  START
      DO 50 JZTAB=1,87                                                   START
      ZTAB = 202-2*JZTAB                                                 START
      IF (ZTAB .GT. Z(JZ3)) GO TO 50                                     START
      Y3SUN(JZ3) = (TABO2(JZTAB-1)-TABO2(JZTAB))*(Z(JZ3)-ZTAB)/2.0       START
     1              + TABO2(JZTAB)                                       START
      Y4SUN(JZ3) = (TABO3(JZTAB-1)-TABO3(JZTAB))*(Z(JZ3)-ZTAB)/2.0       S'.ART
     1             + TABO3(JZTAB)                                        START
      GO TO 55                                                           START
   50 CONTINUE                                                           START
      PRINT 995                                                          START
      IFLAG9 = 1                                                         START
      GO TO 9000                                                         START
   55 IF (IFLAG6 .EQ. 0) YHOLD(3,JZ3) = Y3SUN(JZ3)                       START
      IF (IFLAG7 .EQ. 0) YHOLD(4,JZ3) = Y4SUN(JZ3)                       START
   60 CONTINUE                                                           START
   80 DO 85 M6=5,NSPEC                                                   START
      IF (YNEW(M6) .GT. 1.0) GO TO 85                                    START
      DO 83 JZ6=1,JZEND                                                  START
      YHOLD(M6,JZ6) = 5.0E-6*YHOLD(3,JZ6)                                START
   83 CONTINUE                                                           START
   85 CONTINUE                                                           START
```

```
      DO 110 ITCH=1,10                                                  START
      CHECK = 0.0                                                       START
      DO 100 JZ5=1,JZEND                                                START
      SUMIN(ITCH,JZ5) = 0.0                                             START
      DO 90 M5=3,NSPEC                                                  START
      SUMIN(ITCH,JZ5) = SUMIN(ITCH,JZ5) + YHOLD(M5,JZ5)*IATOM(ITCH,M5)  START
   90 CONTINUE                                                          START
      CHECK = CHECK + SUMIN(ITCH,JZ5)                                   START
  100 CONTINUE                                                          START
      IF (CHECK .GT. 1.0) GO TO 110                                     START
      ITCHXX = ITCH-1                                                   START
      GO TO 120                                                         START
  110 CONTINUE                                                          START
  120 TOLD = T                                                          START
      DO 150 M = 1,NSPEC                                                START
      DO 140 JZ = 1,JZEND                                               START
      YPRINT(M,JZ) = YHOLD(M,JZ)                                        START
      YLAST(M,JZ) = YHOLD(M,JZ)                                         START
  140 CONTINUE                                                          START
  150 CONTINUE                                                          START
  999 FORMAT(I2,2X,10(I1,1X))                                           START
  998 FORMAT (I2)                                                       START
  997 FORMAT (7(E8.2,2X))                                               START
  996 FORMAT (1H ,*NO END OF START PROFILES FOUND.*)                    START
  995 FORMAT (1H ,*THE ALTITUDES ARE OUTSIDE THE BOUNDS OF 30-200 KM SO START
     1YOULL HAVE TO PUT IN YOUR OWN PROFILES FOR O2 AND O3.*)           START
  994 FORMAT (5(E8.2,2X),2(I1,2X),E8.2,2X,I1)                           START
  993 FORMAT (2(E9.2,2X),5(I2,2X))                                      START
 9000 RETURN                                                            START
      END                                                               START -
```

```
      SUBROUTINE ZENANG                                                   ZENANG
      COMMON /A/      A(51)           ,ARK(4)          ,B(51)           , ZENANG
     1 BRK(4)        ,CRK(4)          ,F(20)           ,FF(20)          , ZENANG
     2 FLUX(120)     ,FLXINF(120)     ,FRATE(20)       ,FTERM(40)       , ZENANG
     3 FTERMW(40)    ,G(20)           ,GG(20)          ,HBAR(20)        , ZENANG
     4 H2HOLD(20)    ,H3HOLD(20)      ,IATOM(10,20)    ,IGAIN(4)        , ZENANG
     5 IGAINA(20,20) ,IGAINB(20,20)   ,IGAINC(20,20)   ,IGAIND(20,20)   , ZENANG
     6 ILOSS(4)      ,ILOSSA(20,20)   ,ILOSSB(20,20)   ,ILOSSC(20,20)   , ZENANG
     7 ILOSSD(20,20) ,INFORM(30)      ,JPULL(20)       ,K(1000)         , ZENANG
     8 K0(51)        ,KABS(20,120)    ,KABSN2(120)     ,KABSO2(120)     , ZENANG
     9 KABSO3(120)   ,KRK(80)         ,L1(51)          ,L2(51)          , ZENANG
     X L3(51)        ,LL1(21)         ,LL2(21)         ,LL3(21)         , ZENANG
     1 N(51)         ,NASRAS(1000)    ,NFTERM(20)      ,NTERM(20)       , ZENANG
     2 NUSRUS(400)   ,PHOLD(20,20)    ,PROHLD(71,20)   ,QOLD(20)        , ZENANG
     3 QRK(20)       ,R1(51)          ,R2(51)          ,R3(51)          , ZENANG
     4 RR1(21)       ,RR2(21)         ,RR3(21)         ,SUMIN(10,20)    , ZENANG
     5 TABO2(87)     ,TABO3(87)       ,TEMP(20)        ,TERM(100)       , ZENANG
     6 TERMW(100)    ,Y(20)           ,YHOLD(20,20)    ,YLAST(20,20)    , ZENANG
     7 YNEW(20)      ,YOLD(20)        ,YPRINT(20,20)   ,YSTART(20,20)   , ZENANG
     8 Y3SUN(20)     ,Y4SUN(20)       ,Z(20)           ,ZPRINT(20)        ZENANG
      COMMON /B/   ALPHA , EQUIL , H      , HCOUNT, IFLAG1, IFLAG2,        ZENANG
     2             IFLAG3, IFLAG4, IFLAG5, IFLAG6, IFLAG7, IFLAG8,        ZENANG
     3             IFLAG9, IJZX  , ITCHXX, IUNIT , JCNTRL, JFLAG2,        ZENANG
     4             JFLAG6, JOSHUA, JZEND , JZMAIN, LAT   , LONG  ,        ZENANG
     5             NFREAC, NREAC , NSPEC , PI    , T     , TDAY  ,        ZENANG
     6             THOUR , TMAX1 , TMAX2 , TMIN  , TMONTH, TOLD  ,        ZENANG
     7             TPRINT, TQUIT , TSEC  , ZBOTUM, ZSTEP , ZTOP           ZENANG
      INTEGER HCOUNT,PHOLD,R1,R2,R3,RR1,RR2,RR3,RX1,RX2,RX3,              ZENANG
     1 TMONTH,TDAY,THOUR,TMIN,TSEC,TX                                     ZENANG
      REAL K0,N,KABS,K,KRK,LAT,LONG,K5,K5MAX,KABSO2,KABSO3,KABSN2         ZENANG
      IF (IFLAG3 .NE. 0) GO TO 10                                         ZENANG
      RAD = PI/180.0                                                      ZENANG
      DEC = 23.445*SINF((2.0*PI*(30.4*TMONTH+TDAY-80.0))/365.0)           ZENANG
      IFLAG3 = 1                                                          ZENANG
      PRINT 998                                                           ZENANG
      JCNTRL = 0                                                          ZENANG
      THOLD = TPRINT                                                      ZENANG
      TPRINT = 0                                                          ZENANG
      TZEN = 15.*ABSF(FLOATF(TSEC+60*(TMIN+60*THOUR))-43200.0)/3600.0     ZENANG
      ALPHA = (1.0/RAD)*ACOSF(SINF(LAT*RAD)*SINF(DEC*RAD)                 ZENANG
     1      + COSF(LAT*RAD)*COSF(DEC*RAD)*COSF(TZEN*RAD))                 ZENANG
      CALL HANDLE                                                         ZENANG
      JCNTRL = 1                                                          ZENANG
      TPRINT = THOLD                                                      ZENANG
      GO TO 100                                                           ZENANG
   10 IF (JOSHUA .EQ. 1) GO TO 100                                        ZENANG
      TZEN = 15.*ABSF(FLOATF(TSEC+60*(TMIN+60*THOUR))-43200.0)/3600.0     ZENANG
      ALPHA = (1.0/RAD)*ACOSF(SINF(LAT*RAD)*SINF(DEC*RAD)                 ZENANG
     1      + COSF(LAT*RAD)*COSF(DEC*RAD)*COSF(TZEN*RAD))                 ZENANG
  100 IF (ALPHA .GT. 90.0+SQRTF(Z(1))) GO TO 9000                         ZENANG
      IF (ALPHA .GT. 90.0) GO TO 300                                      ZENANG
      DO 200 JZ1=1,JZEND                                                  ZENANG
      Y3SUN(JZ1) = YHOLD(3,JZ1)                                           ZENANG
      Y4SUN(JZ1) = YHOLD(4,JZ1)                                           ZENANG
```

```
  200 CONTINUE                                                          ZENANG
  300 IF (Y3SUN(2) .GT. Y3SUN(1)) GO TO 310                             ZENANG
      PRINT 997                                                         ZENANG
      IFLAG9 = 1                                                        ZENANG
      GO TO 9000                                                        ZENANG
  310 IF (Y3SUN(1) .GT. 0.0) GO TO 320                                  ZENANG
      H2HOLD(1) = 1.0                                                   ZENANG
      GO TO 330                                                         ZENANG
  320 H2HOLD(1) = (Z(2)-Z(1))/LOGF(Y3SUN(1)/Y3SUN(2))                   ZENANG
  330 IF (Y4SUN(2) .GT. Y4SUN(1)) GO TO 340                             ZENANG
      PRINT 996                                                         ZENANG
      IFLAG9 = 1                                                        ZENANG
      GO TO 9000                                                        ZENANG
  340 IF (Y4SUN(1) .GT. 0.0) GO TO 350                                  ZENANG
      H3HOLD(1) = 1.0                                                   ZENANG
      GO TO 400                                                         ZENANG
  350 H3HOLD(1) = (Z(2)-Z(1))/LOGF(Y4SUN(1)/Y4SUN(2))                   ZENANG
  400 ADD3 = Y3SUN(1)*H2HOLD(1)                                         ZENANG
      ADD4 = Y4SUN(1)*H3HOLD(1)                                         ZENANG
      DO 410 JZ2=2,JZEND                                                ZENANG
      ADD3 = ADD3 + (Y3SUN(JZ2-1)+Y3SUN(JZ2))*(Z(JZ2-1)-Z(JZ2))/2.0     ZENANG
      ADD4 = ADD4 + (Y4SUN(JZ2-1)+Y4SUN(JZ2))*(Z(JZ2-1)-Z(JZ2))/2.0     ZENANG
      H2HOLD(JZ2) = ADD3/Y3SUN(JZ2)                                     ZENANG
      H3HOLD(JZ2) = ADD4/Y4SUN(JZ2)                                     ZENANG
  410 CONTINUE                                                          ZENANG
  997 FORMAT(1H0,*THE TOP SCALE HEIGHT FOR O2 IS NEGATIVE*)             ZENANG
  996 FORMAT (1H0,*THE TOP SCALE HEIGHT FOR O3 IS NEGATIVE*)            ZENANG
  998 FORMAT(1H1,*----------STARTING VALUES----------*,/,               ZENANG
     1 *(REACTION TERM BLOCK CONTAINS RATE COEFFICIENTS*)               ZENANG
 9000 RETURN                                                            ZENANG
      END                                                               ZENANG-
```

```
      SUBROUTINE PRODUC                                                 PRODUC
      COMMON /A/       A(51)             ,ARK(4)           ,B(51)          , PRODUC
     1 BRK(4)          ,CRK(4)           ,F(20)            ,FF(20)         , PRODUC
     2 FLUX(120)       ,FLXINF(120)      ,FRATE(20)        ,FTERM(40)      , PRODUC
     3 FTERMW(40)      ,G(20)            ,GG(20)           ,HBAR(20)       , PRODUC
     4 H2HOLD(20)      ,H3HOLD(20)       ,IATOM(10,20)     ,IGAIN(4)       , PRODUC
     5 IGAINA(20,20)   ,IGAINB(20,20)    ,IGAINC(20,20)    ,IGAIND(20,20)  , PRODUC
     6 ILOSS(4)        ,ILOSSA(20,20)    ,ILOSSB(20,20)    ,ILOSSC(20,20)  , PRODUC
     7 ILOSSD(20,20)   ,INFORM(30)       ,JPULL(20)        ,K(1000)        , PRODUC
     8 K0(51)          ,KABS(20,120)     ,KABSN2(120)      ,KABSO2(120)    , PRODUC
     9 KABSO3(120)     ,KRK(80)          ,L1(51)           ,L2(51)         , PRODUC
     X L3(51)          ,LL1(21)          ,LL2(21)          ,LL3(21)        , PRODUC
     1 N(51)           ,NASRAS(1000)     ,NFTERM(20)       ,NTERM(20)      , PRODUC
     2 NUSRUS(400)     ,PHOLD(20,20)     ,PROHLD(71,20)    ,QOLD(20)       , PRODUC
     3 QRK(20)         ,R1(51)           ,R2(51)           ,R3(51)         , PRODUC
     4 RR1(21)         ,RR2(21)          ,RR3(21)          ,SUMIN(10,20)   , PRODUC
     5 TABO2(87)       ,TABO3(87)        ,TEMP(20)         ,TERM(100)      , PRODUC
     6 TERMW(100)      ,Y(20)            ,YHOLD(20,20)     ,YLAST(20,20)   , PRODUC
     7 YNEW(20)        ,YOLD(20)         ,YPRINT(20,20)    ,YSTART(20,20)  , PRODUC
     8 Y3SUN(20)       ,Y4SUN(20)        ,Z(20)            ,ZPRINT(20)       PRODUC
      COMMON /B/     ALPHA , EQUIL , H      , HCOUNT, IFLAG1, IFLAG2,        PRODUC
     2               IFLAG3, IFLAG4, IFLAG5, IFLAG6, IFLAG7, IFLAG8,         PRODUC
     3               IFLAG9, IJZX  , ITCHXX, IUNIT , JCNTRL, JFLAG2,         PRODUC
     4               JFLAG6, JOSHUA, JZEND , JZMAIN, LAT   , LONG  ,         PRODUC
     5               NFREAC, NREAC , NSPEC , PI    , T     , TDAY  ,         PRODUC
     6               THOUR , TMAX1 , TMAX2 , TMIN  , TMONTH, TOLD  ,         PRODUC
     7               TPRINT, TQUIT , TSEC  , ZBOTUM, ZSTEP , ZTOP            PRODUC
      INTEGER HCOUNT,PHOLD,R1,R2,R3,RR1,RR2,RR3,RX1,RX2,RX3,                 PRODUC
     1 TMONTH,TDAY,THOUR,TMIN,TSEC,TX                                        PRODUC
      REAL K0,N,KABS,K,KRK,LAT,LONG,K5,K5MAX,KABSO2,KABSO3,KABSN2            PRODUC
      CALL Q9EXUN                                                            PRODUC
      IF (NFREAC .EQ. 0) GO TO 9000                                          PRODUC
      ALPHAR = ALPHA*PI/180.0                                                PRODUC
      REARTH = 6371.0                                                        PRODUC
      IF (ALPHA .GT. 90.0) GO TO 100                                         PRODUC
C THIS CALCULATION DONE USING EQN. 53 IN SWIDER, PLANET. SPACE SCI.,         PRODUC
C       VOL. 12, NO. 8, 1964.                                                PRODUC
      PI2 = (PI/2.0)**2                                                      PRODUC
      XX = (REARTH+Z(JZMAIN))/H2HOLD(JZMAIN)                                 PRODUC
      AXLE = (1.0-PI2*(0.115+1.0/LOGF(PI*XX/2.0)))/PI2**2                    PRODUC
      FO2 = EXPF((ALPHAR**2/2.0)/(1.0-ALPHAR**2*(0.115+AXLE*ALPHAR**2)))PRODUC
      XX = (REARTH+Z(JZMAIN))/H3HOLD(JZMAIN)                                 PRODUC
      AXLE = (1.0-PI2*(0.115+1.0/LOGF(PI*XX/2.0)))/PI2**2                    PRODUC
      FO3 = EXPF((ALPHAR**2/2.0)/(1.0-ALPHAR**2*(0.115+AXLE*ALPHAR**2)))PRODUC
      GO TO 300                                                              PRODUC
  100 IF (ALPHA .GT. 90.0+SQRTF(Z(JZMAIN))) GO TO 1000                       PRODUC
      V = (6370.0+Z(JZMAIN))*SINF(ALPHAR)-6370.0                             PRODUC
      IF (V .GE. Z(1)) GO TO 130                                             PRODUC
      IF (V .LE. Z(JZEND)) GO TO 140                                         PRODUC
      DO 110 JZ1=1,JZEND                                                     PRODUC
      IF (V .GT. Z(JZ1)) GO TO 120                                           PRODUC
  110 CONTINUE                                                               PRODUC
      GO TO 140                                                              PRODUC
  120 HO2 = (H2HOLD(JZ1)-H2HOLD(JZ1-1))*(V-Z(JZ1-1))/(Z(JZ1)-Z(JZ1-1))       PRODUC
```

```
     1        +H2HOLD(JZ1-1)                                                    PRODUC
      HO3 = (H3HOLD(JZ1)-H2HOLD(JZ1-1))*(V-Z(JZ1-1))/(Z(JZ1)-Z(JZ1-1))   PRODUC
     1        +H3HOLD(JZ1-1)                                                    PRODUC
      GO TO 150                                                                 PRODUC
  130 HO2 = H2HOLD(JZ1)                                                         PRODUC
      HO3 = H3HOLD(JZ1)                                                         PRODUC
      GO TO 150                                                                 PRODUC
  140 HO2 = H2HOLD(JZEND)                                                       PRODUC
      HO3 = H3HOLD(JZEND)                                                       PRODUC
  150 XO2 = (ALPHA-90.0)/SQRTF(HO2)                                             PRODUC
      XO3 = (ALPHA-90.0)/SQRTF(HO3)                                             PRODUC
      IF (XO2 .GT. 2.5) GO TO 180                                               PRODUC
      ERFO2 = XO2                                                               PRODUC
      FXX = 1.0                                                                 PRODUC
      DO 160 IERF=1,50                                                          PRODUC
      FXX = FXX*IERF                                                            PRODUC
      TXX = (-1)**IERF*XO2**(2*IERF+1)/(FXX*(2*IERF+1))                         PRODUC
      ERFO2 = ERFO2 + TXX                                                       PRODUC
      IF (ABS(TXX) .LT. ABS(ERFO2*1.0E-4)) GO TO 170                            PRODUC
  160 CONTINUE                                                                  PRODUC
  170 ERFO2 = ERFO2*2.0/SQRTF(PI)                                               PRODUC
      GO TO 190                                                                 PRODUC
  180 ERFO2 = 1.00                                                              PRODUC
  190 IF (XO3 .GT. 2.5) GO TO 220                                               PRODUC
      ERFO3 = XO3                                                               PRODUC
      FXX = 1.0                                                                 PRODUC
      DO 200 IERF2=1,50                                                         PRODUC
      FXX = FXX*IERF2                                                           PRODUC
      TXX = (-1)**IERF2*XO3**(2*IERF2+1)/(FXX*(2*IERF2+1))                      PRODUC
      ERFO3 = ERFO3 + TXX                                                       PRODUC
      IF (ABS(TXX) .LT. ABS(ERFO3*1.0E-4)) GO TO 210                            PRODUC
  200 CONTINUE                                                                  PRODUC
  210 ERFO3 = ERFO3*2.0/SQRTF(PI)                                               PRODUC
      GO TO 230                                                                 PRODUC
  220 ERFO3 = 1.0                                                               PRODUC
  230 FO2 = (101.4/SQRTF(HO2))*(1.0+ERFO2)                                      PRODUC
      FO3 = (101.4/SQRTF(HO3))*(1.0+ERFO3)                                      PRODUC
  300 DO 310 LAMDA1=1,120                                                       PRODUC
      FLUX(LAMDA1) = FLXINF(LAMDA1)*EXPF(                                       PRODUC
     1 -KABSO2(LAMDA1)*Y3SUN(JZMAIN)*H2HOLD(JZMAIN)*1.0E5*FO2                   PRODUC
     1 -KABSO3(LAMDA1)*Y4SUN(JZMAIN)*H3HOLD(JZMAIN)*1.0E5*FO3)                  PRODUC
  310 CONTINUE                                                                  PRODUC
      DO 330 II=1,NFREAC                                                        PRODUC
      FRATE(II) = 0.0                                                           PRODUC
      DO 320 LAMDA2=1,120                                                       PRODUC
      FRATE(II) = FRATE(II) + KABS(II,LAMDA2)*FLUX(LAMDA2)                      PRODUC
  320 CONTINUE                                                                  PRODUC
  330 CONTINUE                                                                  PRODUC
      GO TO 9000                                                                PRODUC
 1000 DO 1010 II = 1,NFREAC                                                     PRODUC
      FRATE(II) = 0.0                                                           PRODUC
 1010 CONTINUE                                                                  PRODUC
 9000 CALL R9EXUN                                                               PRODUC
      RETURN                                                                    PRODUC
      END                                                                       PRODUC-
```

```
      SUBROUTINE PHYSICS                                                PHYSICS
      COMMON /A/       A(51)            ,ARK(4)           ,B(51)          , PHYSICS
     1 BRK(4)          ,CRK(4)          ,F(20)            ,FF(20)         , PHYSICS
     2 FLUX(120)       ,FLXINF(120)     ,FRATE(20)        ,FTERM(40)      , PHYSICS
     3 FTERMW(40)      ,G(20)           ,GG(20)           ,HBAR(20)       , PHYSICS
     4 H2HOLD(20)      ,H3HOLD(20)      ,IATOM(10,20)     ,IGAIN(4)       , PHYSICS
     5 IGAINA(20,20)   ,IGAINB(20,20)   ,IGAINC(20,20)    ,IGAIND(20,20)  , PHYSICS
     6 ILOSS(4)        ,ILOSSA(20,20)   ,ILOSSB(20,20)    ,ILOSSC(20,20)  , PHYSICS
     7 ILOSSD(20,20)   ,INFORM(30)      ,JPULL(20)        ,K(1000)        , PHYSICS
     8 KO(51)          ,KABS(20,120)    ,KABSN2(120)      ,KABSO2(120)    , PHYSICS
     9 KABSO3(120)     ,KRK(80)         ,L1(51)           ,L2(51)         , PHYSICS
     X L3(51)          ,LL1(21)         ,LL2(21)          ,LL3(21)        , PHYSICS
     1 N(51)           ,NASRAS(1000)    ,NFTERM(20)       ,NTERM(20)      , PHYSICS
     2 NUSRUS(400)     ,PHOLD(20,20)    ,PROHLD(71,20)    ,QOLD(20)       , PHYSICS
     3 QRK(20)         ,R1(51)          ,R2(51)           ,R3(51)         , PHYSICS
     4 RR1(21)         ,RR2(21)         ,RR3(21)          ,SUMIN(10,20)   , PHYSICS
     5 TABO2(87)       ,TABO3(87)       ,TEMP(20)         ,TERM(100)      , PHYSICS
     6 TERMW(100)      ,Y(20)           ,YHOLD(20,20)     ,YLAST(20,20)   , PHYSICS
     7 YNEW(20)        ,YOLD(20)        ,YPRINT(20,20)    ,YSTART(20,20)  , PHYSICS
     8 Y3SUN(20)       ,Y4SUN(20)       ,Z(20)            ,ZPRINT(20)       PHYSICS
      COMMON /B/  ALPHA , EQUIL , H      , HCOUNT, IFLAG1, IFLAG2,          PHYSICS
     2            IFLAG3, IFLAG4, IFLAG5, IFLAG6, IFLAG7, IFLAG8,           PHYSICS
     3            IFLAG9, IJZX  , ITCHXX, IUNIT , JCNTRL, JFLAG2,           PHYSICS
     4            JFLAG6, JOSHUA, JZEND , JZMAIN, LAT   , LONG  ,           PHYSICS
     5            NFREAC, NREAC , NSPEC , PI    , T     , TDAY  ,           PHYSICS
     6            THOUR , TMAX1 , TMAX2 , TMIN  , TMONTH, TOLD  ,           PHYSICS
     7            TPRINT, TQUIT , TSEC  , ZBOTUM, ZSTEP , ZTOP              PHYSICS
      INTEGER HCOUNT,PHOLD,R1,R2,R3,RR1,RR2,RR3,RX1,RX2,RX3,                PHYSICS
     1 TMONTH,TDAY,THOUR,TMIN,TSEC,TX                                       PHYSICS
      REAL KO,N,KABS,K,KRK,LAT,LONG,K5,K5MAX,KABSO2,KABSO3,KABSN2           PHYSICS
      DO 800 M1=3,NSPEC                                                     PHYSICS
      IIIX = (M1-3)*NREAC                                                   PHYSICS
      IIFX = (M1-3)*NFREAC                                                  PHYSICS
      NNSTOP = NTERM(M1) + NFTERM(M1)                                       PHYSICS
      NTX = NTERM(M1)                                                       PHYSICS
      DO 700 JZ1=1,JZEND                                                    PHYSICS
      DO 100 ILOOK1=1,4                                                     PHYSICS
      ILOSS(ILOOK1) = IGAIN(ILOOK1) = 0                                     PHYSICS
  100 CONTINUE                                                              PHYSICS
      DO 600 ILOOK2=1,4                                                     PHYSICS
      ILMAX = IGMAX = 0                                                     PHYSICS
      TLMAX = TGMAX = 0.0                                                   PHYSICS
      DO 500 NN=1,NNSTOP                                                    PHYSICS
      IF (NN .GT. NTX) GO TO 110                                            PHYSICS
      NX = NASRAS(IIIX+NN)                                                  PHYSICS
      GO TO 150                                                             PHYSICS
  110 NX = NUSRUS(IIFX+NN-NTX)                                              PHYSICS
  150 ILX = ILOOK2-1                                                        PHYSICS
      IF (NN .GT. NTX .AND. NX .GT. NFREAC) GO TO 300                       PHYSICS
      IF (NN .LE. NTX .AND. NX .GT. NREAC) GO TO 300                        PHYSICS
      DO 200 ILOOK3=1,ILX                                                   PHYSICS
      IF (NN .GT. NTX) GO TO 180                                            PHYSICS
      IF (NX .EQ. ILOSS(ILOOK3)) GO TO 500                                  PHYSICS
      GO TO 200                                                             PHYSICS
```

```
180 IF (NX+NREAC .EQ. ILOSS(ILOOK3)) GO TO 500                        PHYSICS
200 CONTINUE                                                            PHYSICS
250 IF (NN .GT. NTX) GO TO 260                                          PHYSICS
    IF (PROHLD(NX,JZ1) .GT. TLMAX) GO TO 270                            PHYSICS
    GO TO 500                                                           PHYSICS
260 IF (PROHLD(NX+NREAC,JZ1) .LE. TLMAX) GO TO 500                      PHYSICS
    TLMAX = PROHLD(NX+NREAC,JZ1)                                        PHYSICS
    ILMAX = NX+NREAC                                                    PHYSICS
    GO TO 500                                                           PHYSICS
270 TLMAX = PROHLD(NX,JZ1)                                              PHYSICS
    ILMAX = NX                                                          PHYSICS
    GO TO 500                                                           PHYSICS
300 IF (NN .GT. NTX) GO TO 310                                          PHYSICS
    NX = NX-NREAC                                                       PHYSICS
    GO TO 320                                                           PHYSICS
310 NX = NX-NFREAC                                                      PHYSICS
320 DO 400 ILOOK4=1,ILX                                                 PHYSICS
    IF (NN .GT. NTX) GO TO 350                                          PHYSICS
    IF (NX .EQ. IGAIN(ILOOK4)) GO TO 500                                PHYSICS
    GO TO 400                                                           PHYSICS
350 IF (NX+NREAC .EQ. IGAIN(ILOOK4)) GO TO 500                          PHYSICS
400 CONTINUE                                                            PHYSICS
    IF (NN .GT. NTX) GO TO 410                                          PHYSICS
    IF (PROHLD(NX,JZ1) .GT. TGMAX) GO TO 420                            PHYSICS
    GO TO 500                                                           PHYSICS
410 IF (PROHLD(NX+NREAC,JZ1) .LE. TGMAX) GO TO 500                      PHYSICS
    TGMAX = PROHLD(NX+NREAC,JZ1)                                        PHYSICS
    IGMAX = NX+NREAC                                                    PHYSICS
    GO TO 500                                                           PHYSICS
420 TGMAX = PROHLD(NX,JZ1)                                              PHYSICS
    IGMAX = NX                                                          PHYSICS
500 CONTINUE                                                            PHYSICS
    IF (ILOOK2 .EQ. 1) GO TO 550                                        PHYSICS
    IF (TLMAX .GE. 0.1*BIGL) ILOSS(ILOOK2) = ILMAX                      PHYSICS
    IF (TGMAX .GE. 0.1*BIGG) IGAIN(ILOOK2) = IGMAX                      PHYSICS
    GO TO 600                                                           PHYSICS
550 ILOSS(1) = ILMAX                                                    PHYSICS
    IGAIN(1) = IGMAX                                                    PHYSICS
    BIGL = TLMAX                                                        PHYSICS
    BIGG = TGMAX                                                        PHYSICS
600 CONTINUE                                                            PHYSICS
    IF (ILOSS(1) .EQ. 0) GO TO 610                                      PHYSICS
    ILOSSA(M1,JZ1) = ILOSS(1)+100                                       PHYSICS
    ILOSSB(M1,JZ1) = ILOSS(2)+100                                       PHYSICS
    ILOSSC(M1,JZ1) = ILOSS(3)+100                                       PHYSICS
    ILOSSD(M1,JZ1) = ILOSS(4)+100                                       PHYSICS
    GO TO 620                                                           PHYSICS
610 ILOSSA(M1,JZ1)=ILOSSB(M1,JZ1)=ILOSSC(M1,JZ1)=ILOSSD(M1,JZ1) = -1   PHYSICS
620 IF (IGAIN(1) .EQ. 0) GO TO 630                                      PHYSICS
    IGAINA(M1,JZ1) = IGAIN(1)+100                                       PHYSICS
    IGAINB(M1,JZ1) = IGAIN(2)+100                                       PHYSICS
    IGAINC(M1,JZ1) = IGAIN(3)+100                                       PHYSICS
    IGAIND(M1,JZ1) = IGAIN(4)+100                                       PHYSICS
    GO TO 700                                                           PHYSICS
630 IGAINA(M1,JZ1)=IGAINB(M1,JZ1)=IGAINC(M1,JZ1)=IGAIND(M1,JZ1) = -1   PHYSICS
700 CONTINUE                                                            PHYSICS
800 CONTINUE                                                            PHYSICS
    RETURN                                                              PHYSICS
    END                                                                 PHYSIC-
```

```
      SUBROUTINE HANDLE                                                 HANDLE
      COMMON /A/       A(51)              ,ARK(4)          ,B(51)       , HANDLE
     1 BRK(4)          ,CRK(4)            ,F(20)           ,FF(20)      , HANDLE
     2 FLUX(120)       ,FLXINF(120)       ,FRATE(20)       ,FTERM(40)   , HANDLE
     3 FTERMW(40)      ,G(20)             ,GG(20)          ,HBAR(20)    , HANDLE
     4 H2HOLD(20)      ,H3HOLD(20)        ,IATOM(10,20)    ,IGAIN(4)    , HANDLE
     5 IGAINA(20,20)   ,IGAINB(20,20)     ,IGAINC(20,20)   ,IGAIND(20,20) , HANDLE
     6 ILOSS(4)        ,ILOSSA(20,20)     ,ILOSSB(20,20)   ,ILOSSC(20,20) , HANDLE
     7 ILOSSD(20,20)   ,INFORM(30)        ,JPULL(20)       ,K(1000)     , HANDLE
     8 K0(51)          ,KABS(20,120)      ,KABSN2(120)     ,KABSO2(120) , HANDLE
     9 KABSO3(120)     ,KRK(80)           ,L1(51)          ,L2(51)      , HANDLE
     X L3(51)          ,LL1(21)           ,LL2(21)         ,LL3(21)     , HANDLE
     1 N(51)           ,NASRAS(1000)      ,NFTERM(20)      ,NTERM(20)   , HANDLE
     2 NUSRUS(400)     ,PHOLD(20,20)      ,PROHLD(71,20)   ,QOLD(20)    , HANDLE
     3 QRK(20)         ,R1(51)            ,R2(51)          ,R3(51)      , HANDLE
     4 RR1(21)         ,RR2(21)           ,RR3(21)         ,SUMIN(10,20) , HANDLE
     5 TABO2(87)       ,TABO3(87)         ,TEMP(20)        ,TERM(100)   , HANDLE
     6 TERMW(100)      ,Y(20)             ,YHOLD(20,20)    ,YLAST(20,20) , HANDLE
     7 YNEW(20)        ,YOLD(20)          ,YPRINT(20,20)   ,YSTART(20,20) , HANDLE
     8 Y3SUN(20)       ,Y4SUN(20)         ,Z(20)           ,ZPRINT(20)    HANDLE
      COMMON /B/   ALPHA , EQUIL , H     , HCOUNT, IFLAG1, IFLAG2,      HANDLE
     2             IFLAG3, IFLAG4, IFLAG5, IFLAG6, IFLAG7, IFLAG8,      HANDLE
     3             IFLAG9, IJZX  , ITCHXX, IUNIT , JCNTRL, JFLAG2,      HANDLE
     4             JFLAG6, JOSHUA, JZEND , JZMAIN, LAT   , LONG  ,      HANDLE
     5             NFREAC, NREAC , NSPEC , PI    , T     , TDAY  ,      HANDLE
     6             THOUR , TMAX1 , TMAX2 , TMIN  , TMONTH, TOLD  ,      HANDLE
     7             TPRINT, TQUIT , TSEC  , ZBOTUM, ZSTEP , ZTOP         HANDLE
      INTEGER HCOUNT,PHOLD,R1,R2,R3,RR1,RR2,RR3,RX1,RX2,RX3,            HANDLE
     1 TMONTH,TDAY,THOUR,TMIN,TSEC,TX                                   HANDLE
      REAL K0,N,KABS,K,KRK,LAT,LONG,K5,K5MAX,KABSO2,KABSO3,KABSN2       HANDLE
      TX=TPRINT+TSEC+TMIN*60+THOUR*3600+TDAY*86400+TMONTH*2592000       HANDLE
      TMONTH = TX/2592000                                               HANDLE
      TX = TX-TMONTH*2592000                                            HANDLE
      TDAY = TX/86400                                                   HANDLE
      TX = TX - TDAY*86400                                              HANDLE
      THOUR = TX/3600                                                   HANDLE
      TX = TX - THOUR*3600                                              HANDLE
      TMIN = TX/60                                                      HANDLE
      TX = TX - TMIN*60                                                 HANDLE
      TSEC = TX                                                         HANDLE
      ITSEC = TSEC + 100                                                HANDLE
      ITMIN = TMIN + 100                                                HANDLE
      ITHOUR = THOUR + 100                                              HANDLE
      ITDAY = TDAY + 100                                                HANDLE
      PRINT 999, JCNTRL,ITHOUR,ITMIN,ITSEC,ITDAY,TMONTH                 HANDLE
      ITIME=KLOCK(1)                                                    HANDLE
      TIME=ITIME/1000.                                                  HANDLE
      PRINT 966,TIME                                                    HANDLE
      PRINT 998, T                                                      HANDLE
      PRINT 997, ALPHA                                                  HANDLE
      PRINT 994                                                         HANDLE
      NTREAC = NREAC + NFREAC                                           HANDLE
      IF (NTREAC .LT. 8) GO TO 10                                       HANDLE
      PRINT 993                                                         HANDLE
```

```
      IF (NTREAC .LT. 16) GO TO 10                                      HANDLE
      PRINT 992                                                         HANDLE
      IF (NTREAC .LT. 24) GO TO 10                                      HANDLE
      PRINT 991                                                         HANDLE
      IF (NTREAC .LT. 32) GO TO 10                                      HANDLE
      PRINT 965                                                         HANDLE
      IF (NTREAC .LT. 40) GO TO 10                                      HANDLE
      PRINT 964                                                         HANDLE
      IF (NTREAC .LT. 48) GO TO 10                                      HANDLE
      PRINT 963                                                         HANDLE
      IF (NTREAC .LT. 56) GO TO 10                                      HANDLE
      PRINT 962                                                         HANDLE
      IF (NTREAC .LT. 64) GO TO 10                                      HANDLE
      PRINT 961                                                         HANDLE
   10 DO 20 JZ = 1,JZEND                                                HANDLE
      PRINT 982, Z(JZ), HBAR(JZ), (PROHLD(I1,JZ),I1=1,7)                HANDLE
      IF (NTREAC .LT. 8) GO TO 20                                       HANDLE
      PRINT 989, (PROHLD(I2,JZ),I2=8,15)                                HANDLE
      IF (NTREAC .LT. 16) GO TO 20                                      HANDLE
      PRINT 988, (PROHLD(I3,JZ),I3=16,23)                               HANDLE
      IF (NTREAC .LT. 24) GO TO 20                                      HANDLE
      PRINT 987, (PROHLD(I4,JZ),I4=24,31)                               HANDLE
      IF (NTREAC .LT. 32) GO TO 20                                      HANDLE
      PRINT 989, (PROHLD(I2,JZ),I2=32,39)                               HANDLE
      IF (NTREAC .LT. 40) GO TO 20                                      HANDLE
      PRINT 988, (PROHLD(I3,JZ),I3=40,47)                               HANDLE
      IF (NTREAC .LT. 48) GO TO 20                                      HANDLE
      PRINT 987, (PROHLD(I4,JZ),I4=48,55)                               HANDLE
      IF (NTREAC .LT. 56) GO TO 20                                      HANDLE
      PRINT 989, (PROHLD(I2,JZ),I2=56,63)                               HANDLE
      IF (NTREAC .LT. 64) GO TO 20                                      HANDLE
      PRINT 988, (PROHLD(I3,JZ),I3=64,72)                               HANDLE
   20 CONTINUE                                                          HANDLE
      PRINT 986                                                         HANDLE
      IF (NSPEC .LT. 10) GO TO 30                                       HANDLE
      PRINT 985                                                         HANDLE
      IF (NSPEC .LT. 18) GO TO 30                                       HANDLE
      PRINT 984                                                         HANDLE
      IF (NSPEC .LT. 26) GO TO 30                                       HANDLE
      PRINT 983                                                         HANDLE
   30 DO 40 JZ2=1,JZEND                                                 HANDLE
      PRINT 990, ZPRINT(JZ2), (YPRINT(I5,JZ2),I5=2,9)                   HANDLE
      IF (NSPEC .LT. 10) GO TO 40                                       HANDLE
      PRINT 989, (YPRINT(I6,JZ2),I6=10,17)                              HANDLE
      IF (NSPEC .LT. 18) GO TO 40                                       HANDLE
      PRINT 988, (YPRINT(I7,JZ2),I7=18,25)                              HANDLE
      IF (NSPEC .LT. 26) GO TO 40                                       HANDLE
      PRINT 987, (YPRINT(I8,JZ2),I8=26,33)                              HANDLE
   40 CONTINUE                                                          HANDLE
      IF (JCNTRL .EQ. 0) GO TO 9000                                     HANDLE
      PRINT 975                                                         HANDLE
      IF (NSPEC .LT. 10) GO TO 50                                       HANDLE
      PRINT 974                                                         HANDLE
      IF (NSPEC .LT. 18) GO TO 50                                       HANDLE
```

```
      PRINT 973                                                         HANDLE
      IF (NSPEC .LT. 26) GO TO 50                                       HANDLE
      PRINT 972                                                         HANDLE
   50 DO 60 JZ3=1,JZEND                                                 HANDLE
      PRINT 971,ZPRINT(JZ3),(ILOSSA(M1,JZ3),ILOSSB(M1,JZ3),             HANDLE
     1  ILOSSC(M1,JZ3),ILOSSD(M1,JZ3),M1=2,9)                           HANDLE
      IF (NSPEC .LT. 10) GO TO 60                                       HANDLE
      PRINT 970, (ILOSSA(M2,JZ3),ILOSSB(M2,JZ3),ILOSSC(M2,JZ3),         HANDLE
     1  ILOSSD(M2,JZ3),M2=10,17)                                        HANDLE
      IF (NSPEC .LT. 18) GO TO 60                                       HANDLE
      PRINT 969, (ILOSSA(M3,JZ3),ILOSSB(M3,JZ3),ILOSSC(M3,JZ3),         HANDLE
     1  ILOSSD(M3,JZ3),M3=18,25)                                        HANDLE
      IF (NSPEC .LT. 26) GO TO 60                                       HANDLE
      PRINT 968, (ILOSSA(M4,JZ3),ILOSSB(M4,JZ3),ILOSSC(M4,JZ3),         HANDLE
     1  ILOSSD(M4,JZ3),M4=26,33)                                        HANDLE
   60 CONTINUE                                                          HANDLE
      PRINT 967                                                         HANDLE
      IF (NSPEC .LT. 10) GO TO 70                                       HANDLE
      PRINT 974                                                         HANDLE
      IF (NSPEC .LT. 18) GO TO 70                                       HANDLE
      PRINT 973                                                         HANDLE
      IF (NSPEC .LT. 26) GO TO 70                                       HANDLE
      PRINT 972                                                         HANDLE
   70 DO 80 JZ4=1,JZEND                                                 HANDLE
      PRINT 971,ZPRINT(JZ4),(IGAINA(M5,JZ4),IGAINB(M5,JZ4),             HANDLE
     1  IGAINC(M5,JZ4),IGAIND(M5,JZ4),M5=2,9)                           HANDLE
      IF (NSPEC .LT. 10) GO TO 80                                       HANDLE
      PRINT 970,(IGAINA(M6,JZ4),IGAINB(M6,JZ4),IGAINC(M6,JZ4),          HANDLE
     1  IGAIND(M6,JZ4),M6=10,17)                                        HANDLE
      IF (NSPEC .LT. 18) GO TO 80                                       HANDLE
      PRINT 969,(IGAINA(M7,JZ4),IGAINB(M7,JZ4),IGAINC(M7,JZ4),          HANDLE
     1  IGAIND(M7,JZ4),M7=18,25)                                        HANDLE
      IF (NSPEC .LT. 26) GO TO 80                                       HANDLE
      PRINT 968,(IGAINA(M8,JZ4),IGAINB(M8,JZ4),IGAINC(M8,JZ4),          HANDLE
     1  IGAIND(M8,JZ4),M8=26,33)                                        HANDLE
   80 CONTINUE                                                          HANDLE
      IF (JZEND .GT. 7) GO TO 90                                        HANDLE
      WRITE (13,899) (M,(YHOLD(M,JZ),JZ=1,7),M=3,NSPEC),99              HANDLE
      GO TO 110                                                         HANDLE
   90 IF (JZEND .GT. 14) GO TO 100                                      HANDLE
      WRITE (13,898) (M,(YHOLD(M,JZ),JZ=1,14),M=3,NSPEC),99             HANDLE
      GO TO 110                                                         HANDLE
  100 WRITE (13,897) (M,(YHOLD(M,JZ),JZ=1,20),M=3,NSPEC),99             HANDLE
  110 CONTINUE                                                          HANDLE
      REWIND 13                                                         HANDLE
  897 FORMAT(I2,/,2(7(E8.2,2X),/),6(E8.2,2X))                           HANDLE
  898 FORMAT(I2,/,7(E8.2,2X),/,7(E8.2,2X))                              HANDLE
  899 FORMAT(I2,/,7(E8.2,2X))                                           HANDLE
  961 FORMAT(1H ,         *              TERM(64)     TERM(65)    TERHANDLE
     1M(66)     TERM(67)     TERM(68)     TERM(69)     TERM(70)    TERMHANDLE
     2(71)*)                                                            HANDLE
  962 FORMAT(1H ,         *             TERM(56)     TERM(57)    TERMHANDLE
     1(58)     TERM(59)     TERM(60)     TERM(61)     TERM(62)    TERM(HANDLE
     263)*)                                                             HANDLE
```

```
963 FORMAT(1H ,          *                          TERM(48)      TERM(49)      TEHANDLE
   1RM(50)      TERM(51)      TERM(52)      TERM(53)      TERM(54)      TERHANDLE
   2M(55)*)                                                                   HANDLE
964 FORMAT(1H ,          *                         TERM(40)      TERM(41)      TERHANDLE
   1M(42)      TERM(43)      TERM(44)      TERM(45)      TERM(46)      TERMHANDLE
   2(47)*)                                                                    HANDLE
965 FORMAT(1H ,          *                        TERM(32)      TERM(33)      TERMHANDLE
   1(34)      TERM(35)      TERM(36)      TERM(37)      TERM(38)      TERM(HANDLE
   239)*)                                                                     HANDLE
966 FORMAT(1X,*COMPUTER TIME= *.F9.3,* SECONDS*)                                HANDLE
967 FORMAT(1H1,*MAJOR PRODUCTION REACTIONS-------*,/,* HEIGHT       NO.HANDLE
   1 2          NO. 3          NO. 4          NO. 5          NO. 6           HANDLE
   2 NO. 7          NO. 8          NO. 9*)                                    HANDLE
968 FORMAT(1H ,8X,8(3X,I2,*/*,2(I2,*,*),I2))                                  HANDLE
969 FORMAT(1H ,7X,8(3X,I2,*/*,2(I2,*,*),I2))                                  HANDLE
970 FORMAT(1H ,6X,8(3X,I2,*/*,2(I2,*,*),I2))                                  HANDLE
971 FORMAT(1H0,F6.1,2X,8(I2,*/*,2(I2,*,*),I2,3X))                              HANDLE
972 FORMAT(1H ,*               NO. 26          NO. 27          NO. 28          HANDLE
   1  NO. 29          NO. 30          NO. 31          NO. 32          NO. 33*)HANDLE
973 FORMAT(1H ,*               NO. 18          NO. 19          NO. 20          HANDLE
   1 NO. 21          NO. 22          NO. 23          NO. 24          NO. 25*) HANDLE
974 FORMAT(1H ,*               NO. 10          NO. 11          NO. 12          HANDLE
   1NO. 13          NO. 14          NO. 15          NO. 16          NO. 17*)  HANDLE
975 FORMAT(1H1,*MAJOR DESTRUCTION REACTIONS--------*,/,* HEIGHT     NO.HANDLE
   1 2          NO. 3          NO. 4          NO. 5          NO. 6           HANDLE
   2 NO. 7          NO. 8          NO. 9*)                                    HANDLE
982 FORMAT (1H0,F6.1,2X,E10.4,3X,7(E10.3,3X))                                 HANDLE
983 FORMAT(1H ,*               Y(26)          Y(27)          Y(28)          YHANDLE
   1(29)          Y(30)          Y(31)          Y(32)          Y(33)*)         HANDLE
984 FORMAT(1H ,*               Y(18)          Y(19)          Y(20)          Y(HANDLE
   121)          Y(22)          Y(23)          Y(24)          Y(25)*)         HANDLE
985 FORMAT(1H ,*               Y(10)          Y(11)          Y(12)          Y(1HANDLE
   13)          Y(14)          Y(15)          Y(16)          Y(17)*)          HANDLE
986 FORMAT(1H1,//,* SPECIES DENSITIES----*,/,* HEIGHT    Y(2)          YHANDLE
   1(3)          Y(4)          Y(5)          Y(6)          Y(7)          Y(HANDLE
   18)          Y(9)*)                                                        HANDLE
987 FORMAT(1H ,12X,8(E10.3,3X))                                               HANDLE
988 FORMAT(1H ,11X,8(E10.3,3X))                                               HANDLE
989 FORMAT(1H ,10X,8(E10.3,3X))                                               HANDLE
990 FORMAT (1H0,F6.1,2X,F11.8,3X,7(E10.3,3X))                                 HANDLE
991 FORMAT(1H ,                *                TERM(24)      TERM(25)      TEHANDLE
   1RM(26)      TERM(27)      TERM(28)      TERM(29)      TERM(30)      TERHANDLE
   2M(31)*)                                                                   HANDLE
992 FORMAT(1H ,                *               TERM(16)      TERM(17)      TERHANDLE
   1M(18)      TERM(19)      TERM(20)      TERM(21)      TERM(22)      TERMHANDLE
   2(23)*)                                                                    HANDLE
993 FORMAT(1H ,*             TERM(8)       TERM(9)       TERM(10)      TERMHANDLE
   1(11)      TERM(12)      TERM(13)      TERM(14)      TERM(15)*)          HANDLE
994 FORMAT(1H0,*REACTION TERMS--------*,/,* HEIGHT    STEP SIZE      TERMHANDLE
   1(1)      TERM(2)      TERM(3)      TERM(4)      TERM(5)      TERM(HANDLE
   26)      TERM(7)*)                                                         HANDLE
997 FORMAT(1H ,*ZENITH ANGLE = *,F6.2,* DEGREES*)                             HANDLE
998 FORMAT(1H ,*RUNNING TIME = *,E10.3,* SECONDS*)                            HANDLE
999 FORMAT(I1,*LOCAL TIME (HOURS,MINUTES,SECONDS) = *,I2,*/*,I2,*/*,I2HANDLE
   1,* ON DAY NUMBER *,I2,*, MONTH NUMBER *,I3)                               HANDLE
9000 RETURN                                                                    HANDLE
    END                                                                       HANDLE-
```

```
      SUBROUTINE RUNKUT                                                 RUNKUT
      COMMON /A/      A(51)             ,ARK(4)           ,B(51)          , RUNKUT
     1 BRK(4)         ,CRK(4)           ,F(20)            ,FF(20)         , RUNKUT
     2 FLUX(120)      ,FLXINF(120)      ,FRATE(20)        ,FTERM(40)      , RUNKUT
     3 FTERMW(40)     ,G(20)            ,GG(20)           ,HBAR(20)       , RUNKUT
     4 H2HOLD(20)     ,H3HOLD(20)       ,IATOM(10,20)     ,IGAIN(4)       , RUNKUT
     5 IGAINA(20,20)  ,IGAINB(20,20)    ,IGAINC(20,20)    ,IGAIND(20,20)  , RUNKUT
     6 ILOSS(4)       ,ILOSSA(20,20)    ,ILOSSB(20,20)    ,ILOSSC(20,20)  , RUNKUT
     7 ILOSSD(20,20)  ,INFORM(30)       ,JPULL(20)        ,K(1000)        , RUNKUT
     8 K0(51)         ,KABS(20,120)     ,KABSN2(120)      ,KABSO2(120)    , RUNKUT
     9 KABSO3(120)    ,KRK(80)          ,L1(51)           ,L2(51)         , RUNKUT
     X L3(51)         ,LL1(21)          ,LL2(21)          ,LL3(21)        , RUNKUT
     1 N(51)          ,NASRAS(1000)     ,NFTERM(20)       ,NTERM(20)      , RUNKUT
     2 NUSRUS(400)    ,PHOLD(20,20)     ,PROHLD(71,20)    ,QOLD(20)       , RUNKUT
     3 QRK(20)        ,R1(51)           ,R2(51)           ,R3(51)         , RUNKUT
     4 RR1(21)        ,RR2(21)          ,RR3(21)          ,SUMIN(10,20)   , RUNKUT
     5 TABO2(87)      ,TABO3(87)        ,TEMP(20)         ,TERM(100)      , RUNKUT
     6 TERMW(100)     ,Y(20)            ,YHOLD(20,20)     ,YLAST(20,20)   , RUNKUT
     7 YNEW(20)       ,YOLD(20)         ,YPRINT(20,20)    ,YSTART(20,20)  , RUNKUT
     8 Y3SUN(20)      ,Y4SUN(20)        ,Z(20)            ,ZPRINT(20)       RUNKUT
      COMMON /B/  ALPHA , EQUIL , H     , HCOUNT, IFLAG1, IFLAG2,           RUNKUT
     2            IFLAG3, IFLAG4, IFLAG5, IFLAG6, IFLAG7, IFLAG8,           RUNKUT
     3            IFLAG9, IJZX  , ITCHXX, IUNIT , JCNTRL, JFLAG2,           RUNKUT
     4            JFLAG6, JOSHUA, JZEND , JZMAIN, LAT   , LONG  ,           RUNKUT
     5            NFREAC, NREAC , NSPEC , PI    , T     , TDAY  ,           RUNKUT
     6            THOUR , TMAX1 , TMAX2 , TMIN  , TMONTH, TOLD  ,           RUNKUT
     7            TPRINT, TQUIT , TSEC  , ZBOTUM, ZSTEP , ZTOP              RUNKUT
      INTEGER HCOUNT,PHOLD,R1,R2,R3,RR1,RR2,RR3,RX1,RX2,RX3,                RUNKUT
     1 TMONTH,TDAY,THOUR,TMIN,TSEC,TX                                       RUNKUT
      REAL K0,N,KABS,K,KRK,LAT,LONG,K5,K5MAX,KABSO2,KABSO3,KABSN2           RUNKUT
      CALL Q9EXUN                                                           RUNKUT
      H = HBAR(JZMAIN)*3.0                                                  RUNKUT
   10 TCHECK = T-TOLD+H                                                     RUNKUT
      IF (TCHECK .LT. TPRINT-1.0E-6) GO TO 20                               RUNKUT
      IF (TCHECK .GT. TPRINT+1.0E-6) H = TPRINT-T+TOLD                      RUNKUT
      IFLAG2 = 1                                                            RUNKUT
   20 DO 30 M1=2,NSPEC                                                      RUNKUT
      YOLD(M1) = Y(M1)                                                      RUNKUT
      QOLD(M1) = QRK(M1)                                                    RUNKUT
   30 CONTINUE                                                              RUNKUT
   40 DO 90 JRK=1,4                                                         RUNKUT
      DO 50 I1=1,NREAC                                                      RUNKUT
      TERM(I1+NREAC) = K(I1*JZEND-IJZX)*Y(L1(I1))*Y(L2(I1))*Y(L3(I1))       RUNKUT
      TERM(I1) =  TERM(I1+NREAC)                                            RUNKUT
   50 CONTINUE                                                              RUNKUT
      DO 52 II1=1,NFREAC                                                    RUNKUT
      FTERM(II1) = FRATE(II1)*Y(LL1(II1))                                   RUNKUT
      FTERM(II1+NFREAC) = FTERM(II1)                                        RUNKUT
   52 CONTINUE                                                              RUNKUT
      JRKX = (JRK-1)*NSPEC                                                  RUNKUT
      DO 70 M2=3,NSPEC                                                      RUNKUT
      MJRK = JRKX+M2                                                        RUNKUT
      IIIX = (M2-3)*NREAC                                                   RUNKUT
      DY = 0.0                                                              RUNKUT
```

```
      NNSTOP = NTERM(M2)                                                  RUNKUT
      DO 60 NN=1,NNSTOP                                                   RUNKUT
      IF (NASRAS(IIIX+NN) .LE. NREAC) GO TO 55                            RUNKUT
      DY = DY+TERM(NASRAS(IIIX+NN))                                       RUNKUT
      GO TO 60                                                            RUNKUT
   55 DY = DY - TERM(NASRAS(IIIX+NN))                                     RUNKUT
   60 CONTINUE                                                            RUNKUT
      IIFX = (M2-3)*NFREAC                                                RUNKUT
      NFSTOP = NFTERM(M2)                                                 RUNKUT
      DO 65 NF=1,NFSTOP                                                   RUNKUT
      IF (NUSRUS(IIFX+NF) .LE. NFREAC) GO TO 63                           RUNKUT
      DY = DY + FTERM(NUSRUS(IIFX+NF))                                    RUNKUT
      GO TO 65                                                            RUNKUT
   63 DY = DY - FTERM(NUSRUS(IIFX+NF))                                    RUNKUT
   65 CONTINUE                                                            RUNKUT
      KRK(MJRK) = DY                                                      RUNKUT
   70 CONTINUE                                                            RUNKUT
      DO 80 M3=3,NSPEC                                                    RUNKUT
      MJRK = JRKX+M3                                                      RUNKUT
      XXRK = ARK(JRK)*(KRK(MJRK)-BRK(JRK)*QRK(M3))                        RUNKUT
      Y(M3) = Y(M3) + H*XXRK                                              RUNKUT
      QRK(M3) = QRK(M3)+3.0*XXRK-CRK(JRK)*KRK(MJRK)                       RUNKUT
   80 CONTINUE                                                            RUNKUT
   90 CONTINUE                                                            RUNKUT
      K5MAX = 0.01                                                        RUNKUT
      DO 100 M4=3,NSPEC                                                   RUNKUT
      IF (Y(M4) .LT. 0.0) Y(M4) = 0.0                                     RUNKUT
      IF (ABS(KRK(M4+NSPEC)) .LT. 1.0E-10 .OR.                            RUNKUT
     1  ABS(1.0-KRK(M4)/KRK(NSPEC+M4)) .LT. 1.0E-2 .OR.                   RUNKUT
     2  ABS(KRK(M4)+KRK(NSPEC+M4)+KRK(2*NSPEC+M4)+KRK(3*NSPEC+M4))*H      RUNKUT
     3  .LT. YOLD(M4)*1.0E-4) GO TO 100                                   RUNKUT
      K5 = ABS((KRK(NSPEC+M4)-KRK(2*NSPEC+M4))/(KRK(M4)-KRK(NSPEC+M4)))   RUNKUT
      IF (K5 .GT. K5MAX) K5MAX = K5                                       RUNKUT
  100 CONTINUE                                                            RUNKUT
      IF (K5MAX .LT. 0.10) GO TO 120                                      RUNKUT
      H = H*0.08/K5MAX                                                    RUNKUT
      IFLAG2 = 0                                                          RUNKUT
      DO 110 M5=2,NSPEC                                                   RUNKUT
      Y(M5) = YOLD(M5)                                                    RUNKUT
      QRK(M5) = QOLD(M5)                                                  RUNKUT
  110 CONTINUE                                                            RUNKUT
      GO TO 40                                                            RUNKUT
  120 T = T+H                                                             RUNKUT
      H = H*0.08/K5MAX                                                    RUNKUT
      HCOUNT = HCOUNT+1                                                   RUNKUT
      IF (IFLAG2 .EQ. 0) GO TO 10                                         RUNKUT
      CALL R9EXUN                                                         RUNKUT
      RETURN                                                              RUNKUT
      END                                                                 RUNKUT-
```

```
      SUBROUTINE WEIRD                                                  WEIRD
      COMMON /A/       A(51)            ,ARK(4)          ,B(51)         , WEIRD
     1 BRK(4)          ,CRK(4)          ,F(20)           ,FF(20)        , WEIRD
     2 FLUX(120)       ,FLXINF(120)     ,FRATE(20)       ,FTERM(40)     , WEIRD
     3 FTERMW(40)      ,G(20)           ,GG(20)          ,HBAR(20)      , WEIRD
     4 H2HOLD(20)      ,H3HOLD(20)      ,IATOM(10,20)    ,IGAIN(4)      , WEIRD
     5 IGAINA(20,20)   ,IGAINB(20,20)   ,IGAINC(20,20)   ,IGAIND(20,20) , WEIRD
     6 ILOSS(4)        ,ILOSSA(20,20)   ,ILOSSB(20,20)   ,ILOSSC(20,20) , WEIRD
     7 ILOSSD(20,20)   ,INFORM(30)      ,JPULL(20)       ,K(1000)       , WEIRD
     8 KO(51)          ,KABS(20,120)    ,KABSN2(120)     ,KABSO2(120)   , WEIRD
     9 KABSO3(120)     ,KRK(80)         ,L1(51)          ,L2(51)        , WEIRD
     X L3(51)          ,LL1(21)         ,LL2(21)         ,LL3(21)       , WEIRD
     1 N(51)           ,NASRAS(1000)    ,NFTERM(20)      ,NTERM(20)     , WEIRD
     2 NUSRUS(400)     ,PHOLD(20,20)    ,PROHLD(71,20)   ,QOLD(20)      , WEIRD
     3 QRK(20)         ,R1(51)          ,R2(51)          ,R3(51)        , WEIRD
     4 RR1(21)         ,RR2(21)         ,RR3(21)         ,SUMIN(10,20)  , WEIRD
     5 TABO2(87)       ,TABO3(87)       ,TEMP(20)        ,TERM(100)     , WEIRD
     6 TERMW(100)      ,Y(20)           ,YHOLD(20,20)    ,YLAST(20,20)  , WEIRD
     7 YNEW(20)        ,YOLD(20)        ,YPRINT(20,20)   ,YSTART(20,20) , WEIRD
     8 Y3SUN(20)       ,Y4SUN(20)       ,Z(20)           ,ZPRINT(20)      WEIRD
      COMMON /B/   ALPHA , EQUIL , H      , HCOUNT, IFLAG1, IFLAG2,       WEIRD
     2             IFLAG3, IFLAG4, IFLAG5, IFLAG6, IFLAG7, IFLAG8,        WEIRD
     3             IFLAG9, IJZX  , ITCHXX, IUNIT , JCNTRL, JFLAG2,        WEIRD
     4             JFLAG6, JOSHUA, JZEND , JZMAIN, LAT   , LONG  ,        WEIRD
     5             NFREAC, NREAC , NSPEC , PI    , T     , TDAY  ,        WEIRD
     6             THOUR , TMAX1 , TMAX2 , TMIN  , TMONTH, TOLD  ,        WEIRD
     7             TPRINT, TQUIT , TSEC  , ZBOTUM, ZSTEP , ZTOP           WEIRD
      INTEGER HCOUNT,PHOLD,R1,R2,R3,RR1,RR2,RR3,RX1,RX2,RX3,              WEIRD
     1 TMONTH,TDAY,THOUR,TMIN,TSEC,TX                                     WEIRD
      REAL KO,N,KABS,K,KRK,LAT,LONG,K5,K5MAX,KABSO2,KABSO3,KABSN2         WEIRD
      AKRACY = 1.0E-3                                                     WEIRD
      HMIN = 1.0E-6                                                       WEIRD
      YNEW(1) = 1.0                                                       WEIRD
      CALL Q9EXUN                                                         WEIRD
      H = HBAR(JZMAIN)*3.0                                                WEIRD
      IF (H .LT. TPRINT-(T-TOLD)) GO TO 10                                WEIRD
      H = TPRINT-(T-TOLD)                                                 WEIRD
      IFLAG2 = 1                                                          WEIRD
   10 DO 100 I1=1,NREAC                                                   WEIRD
      TERM(I1) = K(I1*JZEND-IJZX)*Y(L1(I1))*Y(L2(I1))*Y(L3(I1))           WEIRD
      TERM(I1+NREAC) = TERM(I1)                                           WEIRD
  100 CONTINUE                                                            WEIRD
      DO 200 II1=1,NFREAC                                                 WEIRD
      FTERM(II1) = FRATE(II1)*Y(LL1(II1))                                 WEIRD
      FTERM(II1+NFREAC) = FTERM(II1)                                      WEIRD
  200 CONTINUE                                                            WEIRD
      DO 500 M1=3,NSPEC                                                   WEIRD
      F(M1) = 0.0                                                         WEIRD
      G(M1) = 0.0                                                         WEIRD
      NNSTOP = NTERM(M1)                                                  WEIRD
      IIIX = (M1-3)*NREAC                                                 WEIRD
      DO 300 NN=1,NNSTOP                                                  WEIRD
      NASX = NASRAS(IIIX+NN)                                              WEIRD
      IF (NASX .GT. NREAC) GO TO 250                                      WEIRD
```

```
      G(M1) = G(M1) + TERM(NASX)                                          WEIRD
      GO TO 300                                                           WEIRD
  250 F(M1) = F(M1) + TERM(NASX)                                          WEIRD
  300 CONTINUE                                                            WEIRD
      IIFX = (M1-3)*NFREAC                                                WEIRD
      NFSTOP = NFTERM(M1)                                                 WEIRD
      DO 400 NF=1,NFSTOP                                                  WEIRD
      NUSX = NUSRUS(IIFX+NF)                                              WEIRD
      IF (NUSX .GT. NFREAC) GO TO 350                                     WEIRD
      G(M1) = G(M1) + FTERM(NUSX)                                         WEIRD
      GO TO 400                                                           WEIRD
  350 F(M1) = F(M1) + FTERM(NUSX)                                         WEIRD
  400 CONTINUE                                                            WEIRD
      IF (Y(M1) .LT. 1.0E-100) GO TO 450                                  WEIRD
      G(M1) = G(M1)/Y(M1)                                                 WEIRD
      GO TO 500                                                           WEIRD
  450 G(M1) = 0.0                                                         WEIRD
  500 CONTINUE                                                            WEIRD
  600 DO 700 M2=3,NSPEC                                                   WEIRD
      IF (G(M2) .LT. 1.0E-100 .AND. F(M2) .LT. 1.0E-100) GO TO 675        WEIRD
      IF (G(M2)*H .LT. 1.0E-8) GO TO 675                                  WEIRD
      IF (G(M2)*Y(M2)*1.0E8 .LT. F(M2)) GO TO 650                         WEIRD
      IF (F(M2)*1.0E8 .LT. G(M2)*Y(M2)) GO TO 625                         WEIRD
      YNEW(M2) = (Y(M2)-F(M2)/G(M2))*EXPF(-G(M2)*H) + F(M2)/G(M2)         WEIRD
      GO TO 700                                                           WEIRD
  625 YNEW(M2) = Y(M2)*EXPF(-G(M2)*H)                                     WEIRD
      GO TO 700                                                           WEIRD
  650 YNEW(M2) = Y(M2) + F(M2)*H                                          WEIRD
      GO TO 700                                                           WEIRD
  675 YNEW(M2) = Y(M2)                                                    WEIRD
  700 CONTINUE                                                            WEIRD
      DO 800 I2=1,NREAC                                                   WEIRD
      TERMW(I2)=K(I2*JZEND-IJZX)*YNEW(L1(I2))*YNEW(L2(I2))*YNEW(L3(I2))   WEIRD
      TERMW(I2+NREAC) = TERMW(I2)                                         WEIRD
  800 CONTINUE                                                            WEIRD
      DO 900 II2=1,NFREAC                                                 WEIRD
      FTERMW(II2) = FRATE(II2)*YNEW(LL1(II2))                             WEIRD
      FTERMW(II2+NFREAC) = FTERMW(II2)                                    WEIRD
  900 CONTINUE                                                            WEIRD
      DO 1200 M3=3,NSPEC                                                  WEIRD
      FF(M3) = 0.0                                                        WEIRD
      GG(M3) = 0.0                                                        WEIRD
      NNSTOP = NTERM(M3)                                                  WEIRD
      IIIX = (M3-3)*NREAC                                                 WEIRD
      DO 1000 NN2=1,NNSTOP                                                WEIRD
      NASX = NASRAS(IIIX+NN2)                                             WEIRD
      IF (NASX .GT. NREAC) GO TO 950                                      WEIRD
      GG(M3) = GG(M3) + TERMW(NASX)                                       WEIRD
      GO TO 1000                                                          WEIRD
  950 FF(M3) = FF(M3) + TERMW(NASX)                                       WEIRD
 1000 CONTINUE                                                            WEIRD
      IIFX = (M3-3)*NFREAC                                                WEIRD
      NFSTOP = NFTERM(M3)                                                 WEIRD
      DO 1100 NF2=1,NFSTOP                                                WEIRD
```

```
      NUSX = NUSRUS(IIFX+NF2)                                           WEIRD
      IF (NUSX .GT. NFREAC) GO TO 1050                                  WEIRD
      GG(M3) = GG(M3) + FTERMW(NUSX)                                    WEIRD
      GO TO 1100                                                        WEIRD
 1050 FF(M3) = FF(M3) + FTERMW(NUSX)                                    WEIRD
 1100 CONTINUE                                                          WEIRD
      IF (YNEW(M3) .LT. 1.0E-100) GO TO 1150                            WEIRD
      GG(M3) = GG(M3)/YNEW(M3)                                          WEIRD
      GO TO 1200                                                        WEIRD
 1150 GG(M3) = 0.0                                                      WEIRD
 1200 CONTINUE                                                          WEIRD
      DELMAX = 0.0                                                      WEIRD
      DO 1400 M4=3,NSPEC                                                WEIRD
      IF (Y(M4) .LT. 1.0E-100) GO TO 1400                               WEIRD
      IF (ABS(Y(M4)*G(M4)-F(M4))*EXPF(-G(M4)*H)*1.0E2 .LT. F(M4))       WEIRD
     1             GO TO 1400                                           WEIRD
      IF (GG(M4)*H .GT. 300.0) GO TO 1400                               WEIRD
      IF (GG(M4) .LT. 1.0E-100 .AND. FF(M4) .LT. 1.0E-100) GO TO 1275   WEIRD
      IF (GG(M4)*H .LT. 1.0E-8) GO TO 1275                              WEIRD
      IF (GG(M4)*YNEW(M4)*1.0E8 .LT. FF(M4)) GO TO 1250                 WEIRD
      IF (FF(M4)*1.0E8 .LT. GG(M4)*YNEW(M4)) GO TO 1225                 WEIRD
      YBACK = (YNEW(M4)-FF(M4)/GG(M4))*EXPF(GG(M4)*H)+FF(M4)/GG(M4)     WEIRD
      GO TO 1300                                                        WEIRD
 1225 YBACK = YNEW(M4)*EXPF(GG(M4)*H)                                   WEIRD
      GO TO 1300                                                        WEIRD
 1250 YBACK = YNEW(M4) - FF(M4)*H                                       WEIRD
      GO TO 1300                                                        WEIRD
 1275 YBACK = YNEW(M4)                                                  WEIRD
 1300 DEL = ABS(YBACK/Y(M4) - 1.0)                                      WEIRD
      IF (DEL .GT. DELMAX) DELMAX = DEL                                 WEIRD
 1400 CONTINUE                                                          WEIRD
      IF (DELMAX .LE. AKRACY) GO TO 1500                                WEIRD
      IF (H .LT. 1.01*HMIN) GO TO 1500                                  WEIRD
      H = H/2.0                                                         WEIRD
      IF (H .LT. HMIN) H = HMIN                                         WEIRD
      IFLAG2 = 0                                                        WEIRD
      GO TO 600                                                         WEIRD
 1500 DO 1700 M6=3,NSPEC                                                WEIRD
      FBAR = (F(M6)+FF(M6))/2.0                                         WEIRD
      GBAR = (G(M6)+GG(M6))/2.0                                         WEIRD
      IF (GBAR .LT. 1.0E-100 .AND. FBAR .LT. 1.0E-100) GO TO 1575       WEIRD
      IF (GBAR*H .LT. 1.0E-8) GO TO 1575                                WEIRD
      IF (GBAR*Y(M6)*1.0E8 .LT. FBAR) GO TO 1550                        WEIRD
      IF (FBAR*1.0E8 .LT. GBAR*Y(M6)) GO TO 1525                        WEIRD
      Y(M6) = (Y(M6)-FBAR/GBAR)*EXPF(-GBAR*H) + FBAR/GBAR               WEIRD
      GO TO 1700                                                        WEIRD
 1525 Y(M6) = Y(M6)*EXPF(-GBAR*H)                                       WEIRD
      GO TO 1700                                                        WEIRD
 1550 Y(M6) = Y(M6) + FBAR*H                                            WEIRD
      GO TO 1700                                                        WEIRD
 1575 Y(M6) = Y(M6)                                                     WEIRD
 1700 CONTINUE                                                          WEIRD
      T = T+H                                                           WEIRD
      HCOUNT = HCOUNT + 1                                               WEIRD
```

```
      IF (IFLAG2 .EQ. 1) GO TO 9000                                     WEIRD
      IF (DELMAX .GT. 1.0E-8) GO TO 1800                                WEIRD
      H = H*1.0E2                                                       WEIRD
      GO TO 1900                                                        WEIRD
 1800 H = H*1.5                                                         WEIRD
      IF (H .LT. HMIN) H = HMIN                                         WEIRD
 1900 IF (H .LT. TPRINT - (T-TOLD)) GO TO 10                            WEIRD
      H = TPRINT - (T-TOLD)                                             WEIRD
      IFLAG2 = 1                                                        WEIRD
      GO TO 10                                                          WEIRD
 9000 CALL R9EXUN                                                       WEIRD
      RETURN                                                            WEIRD
      END                                                               WEIRD
         SCOPE
 LOAD
```

```
      SUBROUTINE MICROPLT                                                       MICROPLT
      COMMON /A/       A(51)              ,ARK(4)           ,B(51)            , MICROPLT
     1 BRK(4)          ,CRK(4)            ,F(20)            ,FF(20)           , MICROPLT
     2 FLUX(120)       ,FLXINF(120)       ,FRATE(20)        ,FTERM(40)        , MICROPLT
     3 FTERMW(40)      ,G(20)             ,GG(20)           ,HBAR(20)         , MICROPLT
     4 H2HOLD(20)      ,H3HOLD(20)        ,IATOM(10,20)     ,IGAIN(4)         , MICROPLT
     5 IGAINA(20,20)   ,IGAINB(20,20)     ,IGAINC(20,20)    ,IGAIND(20,20)    , MICROPLT
     6 ILOSS(4)        ,ILOSSA(20,20)     ,ILOSSB(20,20)    ,ILOSSC(20,20)    , MICROPLT
     7 ILOSSD(20,20)   ,INFORM(30)        ,JPULL(20)        ,K(1000)          , MICROPLT
     8 KO(51)          ,KABS(20,120)      ,KABSN2(120)      ,KABSO2(120)      , MICROPLT
     9 KABSO3(120)     ,KRK(80)           ,L1(51)           ,L2(51)           , MICROPLT
     X L3(51)          ,LL1(21)           ,LL2(21)          ,LL3(21)          , MICROPLT
     1 N(51)           ,NASRAS(1000)      ,NFTERM(20)       ,NTERM(20)        , MICROPLT
     2 NUSRUS(400)     ,PHOLD(20,20)      ,PROHLD(71,20)    ,QOLD(20)         , MICROPLT
     3 QRK(20)         ,R1(51)            ,R2(51)           ,R3(51)           , MICROPLT
     4 RR1(21)         ,RR2(21)           ,RR3(21)          ,SUMIN(10,20)     , MICROPLT
     5 TABO2(87)       ,TABO3(87)         ,TEMP(20)         ,TERM(100)        , MICROPLT
     6 TERMW(100)      ,Y(20)             ,YHOLD(20,20)     ,YLAST(20,20)     , MICROPLT
     7 YNEW(20)        ,YOLD(20)          ,YPRINT(20,20)    ,YSTART(20,20)    , MICROPLT
     8 Y3SUN(20)       ,Y4SUN(20)         ,Z(20)            ,ZPRINT(20)         MICROPLT
      COMMON /B/  ALPHA , EQUIL , H      , HCOUNT, IFLAG1, IFLAG2,              MICROPLT
     2            IFLAG3, IFLAG4, IFLAG5, IFLAG6, IFLAG7, IFLAG8,               MICROPLT
     3            IFLAG9, IJZX  , ITCHXX, IUNIT , JCNTRL, JFLAG2,               MICROPLT
     4            JFLAG6, JOSHUA, JZEND , JZMAIN, LAT   , LONG  ,               MICROPLT
     5            NFREAC, NREAC , NSPEC , PI    , T     , TDAY  ,               MICROPLT
     6            THOUR , TMAX1 , TMAX2 , TMIN  , TMONTH, TOLD  ,               MICROPLT
     7            TPRINT, TQUIT , TSEC  , ZBOTUM, ZSTEP , ZTOP                  MICROPLT
      INTEGER HCOUNT,PHOLD,R1,R2,R3,RR1,RR2,RR3,RX1,RX2,RX3,                    MICROPLT
     1 TMONTH,TDAY,THOUR,TMIN,TSEC,TX                                           MICROPLT
      COMMON/DD/IN,IOR,IT,IS,IC,ICC,IX,IY                                       MICROPLT
      COMMON/DDID/ID(4),IFL,LU                                                  MICROPLT
      DATA(ID=32HG W ADAMS X-3918,MICROPLT       )                              MICROPLT
      DIMENSION TEXT1(10),TEXTV(2),IXHOLD(20,20)                                MICROPLT
      DATA(TEXTV=16H HEIGHT(KM)      ),(NCYCLE=0)                               MICROPLT
      IFL = IFL+1                                                               MICROPLT
      MINY=20*(INT(Z(JZEND))/20)                                                MICROPLT
      MAXY=20*(INT(Z(1)-1)/20+1)                                                MICROPLT
      NTICS=(MAXY-MINY)/10+1                                                    MICROPLT
      IF(NCYCLE) GO TO 20                                                       MICROPLT
      NCYCLE=1  $  MAXPOWR=0                                                    MICROPLT
                                                                                MICROPLT
C LOOP 100 COMPUTES RANGE OF YHOLD.                                             MICROPLT
      DO 100 ISPEC=3,NSPEC                                                      MICROPLT
        J=-1                                                                    MICROPLT
10      J=J+1  $  IF(YHOLD(ISPEC,JZEND)/10.**J.GT.1) GO TO 10                   MICROPLT
100   IF(J.GT.MAXPOWR) MAXPOWR=J                                                MICROPLT
                                                                                MICROPLT
C LOOP 800 PLOTS UP TO 5 CURVES/FRAME                                           MICROPLT
20    NOPLTS =(NSPEC+2)/5                                                       MICROPLT
      DO 800 IPLOT=1,NOPLTS                                                     MICROPLT
        IN=IS=1                                                                 MICROPLT
        CALL DDBOX(70,1020,70,1020)                                             MICROPLT
                                                                                MICROPLT
C LOOP 200 DRAWS VERTICAL TICS AND HEIGHTS.                                     MICROPLT
```

```
      DO 200 J=1,NTICS                                                   MICROPLT
        IY=70+950*(J-1)/(NTICS-1)                                        MICROPLT
        IF(MOD(J,2).EQ.0) GO TO 25                                       MICROPLT
        LABEL=MINY+10*(J-1)  $  ENCODE(8,920,TEXT) LABEL                 MICROPLT
920     FORMAT(I4,4X)                                                    MICROPLT
        IX=16  $  CALL DDTAB  $  CALL DDTABNA8(1,TEXT,1)                 MICROPLT
        IF(J.EQ.1.OR.J.EQ.NTICS) GO TO 200                               MICROPLT
25      IX=70  $  ICC=1R-  $  CALL DDTAB  $  CALL DDTABCC                MICROPLT
        IX=1020  $  ICC=1R-  $  CALL DDTAB  $  CALL DDTABCC              MICROPLT
200   CONTINUE   VERTICAL LABELS  AND TIC MARKS                          MICROPLT
                                                                         MICROPLT
C DRAW VERTICAL LINES AND LABEL GRAPH.                                   MICROPLT
    IN=0  $  CALL DDSEGVEC                                               MICROPLT
      LIMIT=MAXPOWR-1                                                    MICROPLT
      DO 300 M=1,LIMIT                                                   MICROPLT
      IX=70+950*M/MAXPOWR  $  IY=70  $  CALL DDXY $  IY=1020             MICROPLT
300   CALL DDXY                                                          MICROPLT
    IN=IOR=1  $  IS=2  $  IX=0  $  IY=460  $  CALL DDTAB                 MICROPLT
    CALL DDTABNA8(2,TEXTV,1)  $  IOR=0                                   MICROPLT
    LABEL=10  $  ENCODE(8,920,TEXT) LABEL                                MICROPLT
      DO 400 M=1,LIMIT                                                   MICROPLT
      IX=36+950*M/MAXPOWR  $  IY=45  $  IS=1  $  CALL DDTAB               MICROPLT
      CALL DDTABNA8(1,TEXT,1)                                            MICROPLT
      ENCODE(8,920,TEXT1) M                                              MICROPLT
      IX=IX+20 + M/10*8  $  IY=55  $  IS=0  $  CALL DDTAB                MICROPLT
400   CALL DDTABNA8(1,TEXT1,1)                                           MICROPLT
    ENCODE(48,940,TEXT1)ALPHA,THOUR,TMIN,TSEC,TMONTH,TDAY                MICROPLT
940 FORMAT(*ALPHA=*F5.1,6X,3I3,*   MONTH*I3*   DAY*I3)                   MICROPLT
    IY=3  $  IX=150  $  IS=2  $  CALL DDTAB                              MICROPLT
    CALL DDTABNA8(6,TEXT1,1)                                             MICROPLT
    IX=470  $  ICC=12B  $  CALL DDTAB  $  CALL DDTABCC                   MICROPLT
    IX=518  $  ICC=12B  $  CALL DDTAB  $  CALL DDTABCC                   MICROPLT
                                                                         MICROPLT
C LOOP 500 SCALES LOG(YHOLD) TO PLOTTER UNITS.                           MICROPLT
      DO 500 J=1,400                                                     MICROPLT
      V=YHOLD(J)  $  IF(V.LT.1.) GO TO 500                               MICROPLT
      IXHOLD(J)=ALOG10(V)*950./MAXPOWR + 70.                             MICROPLT
500   CONTINUE                                                           MICROPLT
                                                                         MICROPLT
C LOOP 600(I) PLOTS EACH CURVE.                                          MICROPLT
    IS=IN=0                                                              MICROPLT
    LIMIT=MIN0(5,NSPEC+3-5*IPLOT)                                        MICROPLT
      DO 600 I=1,LIMIT                                                   MICROPLT
      IND=5*IPLOT+I-3                                                    MICROPLT
      ENCODE(8,960,TEXT) IND                                             MICROPLT
960   FORMAT(I2,6X)                                                      MICROPLT
      IXTEMP=IXHOLD(IND,1) $ IYTEMP=(Z(1)-MINY)*950/(MAXY-MINY)+70.      MICROPLT
                                                                         MICROPLT
C LOOP 600(J) PLOTS SPECIES SYMBOL AT EACH POINT AND DRAWS LINES         MICROPLT
C BETWEEN SYMBOLS.                                                       MICROPLT
        DO 600 J=1,JZEND                                                 MICROPLT
        IX=IXTEMP-8+IND/10*4  $  IY=IYTEMP                               MICROPLT
        CALL DDTAB  $  CALL DDTABNA8(1,TEXT,1)                           MICROPLT
        IX=IXTEMP                                                        MICROPLT
```

```
      IF(J.EQ.JZEND) GO TO 600                                          MICROPLT
      CALL DDCONVEC                                                     MICROPLT
      IXTEMP=IXHOLD(IND,J+1)                                            MICROPLT
      IYTEMP=(Z(J+1)-MINY)*950./(MAXY-MINY) + 70.                       MICROPLT
      XDIF=IXTEMP-IX  $  YDIF=IYTEMP-IY                                 MICROPLT
      D=SQRTF(XDIF*XDIF+YDIF*YDIF)                                      MICROPLT
      IDX=8*XDIF/D  $  IDY=8*YDIF/D                                     MICROPLT
      IX=IX+IDX  $  IY=IY+IDY  $  CALL DDXY                             MICROPLT
      IX=IXTEMP-IDX  $  IY=IYTEMP-IDY $  CALL DDXY                      MICROPLT
600     CONTINUE                                                        MICROPLT
800   CALL DDFRAME                                                      MICROPLT
      RETURN                                                            MICROPLT
      END                                                               MICROP-
```


GPO 830-483